

Oracle® Database

Reference

10g Release 1 (10.1)

Part No. B10755-01

December 2003

Oracle Database Reference, 10g Release 1 (10.1)

Part No. B10755-01

Copyright © 2002, 2003, Oracle. All rights reserved.

Primary Author: Tony Morales

Contributors: Nipun Agarwal, Rick Anderson, Mark Bauer, Ruth Baylis, Neerja Bhatt, Tudor Bosman, George Candea, Wilson Chan, Sumanta Chatterjee, Eugene Chong, Connie Dialeris Green, Harvey Eneman, Bruce Ernst, Ira Greenberg, Steve Harris, Thuvan Hoang, Lilian Hobbs, Namit Jain, Hakkan Jakobsson, Bob Jenkins, Mark Johnson, Jonathan Klein, Sushil Kumar, Tirthankar Lahiri, Bill Lee, Yunrui Li, Juan Loiza, Rich Long, Diana Lorentz, Catherine Luu, Neil MacNaughton, Shankar Raman, Ari Mozes, Gopal Mulagund, Subramanian Muralidhar, Ravi Murthy, Karuna Muthiah, Arvind Nithrakashyap, Ronald Obermarck, Kant Patel, Christopher Racicot, Mark Ramacher, Suvendu Ray, Ann Rhee, Vivian Schupmann, Debbie Steiner, Seema Sundara, Juan Tellez, Alex Tsukerman, Kothanda Umamageswaran, Randy Urbano, Steve Vivian, Eric Voss, Tak Wang, Steve Wertheimer, Andy Witkowski, Daniel Wong, Graham Wood, Brian Wright, Aravind Yalamanchi, Qin Yu, Mohamed Ziauddin

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	xxxix
Preface	xli
Intended Audience	xli
Documentation Accessibility	xli
Structure	xlii
Related Documents	xliii
Conventions	xliv
What's New in Oracle Database Reference?	xlix
Oracle Database 10g Release 1 (10.1) New Features	xlix
Part I Initialization Parameters	
1 Initialization Parameters	
Uses of Initialization Parameters	1-1
Types of Initialization Parameters	1-2
Basic Initialization Parameters	1-3
Parameter Files	1-3
Server Parameter Files	1-3
Initialization Parameter Files	1-4
Changing Parameter Values in a Parameter File	1-6
Parameters by Functional Category	1-6
Modifiable Parameters	1-12
Displaying Current Parameter Values	1-15
Parameters You Should Not Specify in the Parameter File	1-16
When Parameters Are Set Incorrectly	1-16
Reading the Parameter Descriptions	1-16
Initialization Parameter Descriptions	1-17
ACTIVE_INSTANCE_COUNT	1-17
AQ_TM_PROCESSES	1-18
ARCHIVE_LAG_TARGET	1-18
ASM_DISKGROUPS	1-19
ASM_DISKSTRING	1-20
ASM_POWER_LIMIT	1-20

AUDIT_FILE_DEST.....	1-21
AUDIT_SYS_OPERATIONS	1-21
AUDIT_TRAIL	1-21
BACKGROUND_CORE_DUMP	1-22
BACKGROUND_DUMP_DEST	1-22
BACKUP_TAPE_IO_SLAVES.....	1-23
BITMAP_MERGE_AREA_SIZE.....	1-24
BLANK_TRIMMING	1-24
CIRCUITS.....	1-25
CLUSTER_DATABASE.....	1-25
CLUSTER_DATABASE_INSTANCES	1-25
CLUSTER_INTERCONNECTS.....	1-26
COMMIT_POINT_STRENGTH.....	1-27
COMPATIBLE	1-27
CONTROL_FILE_RECORD_KEEP_TIME	1-28
CONTROL_FILES	1-28
CORE_DUMP_DEST.....	1-29
CPU_COUNT	1-29
CREATE_BITMAP_AREA_SIZE.....	1-30
CREATE_STORED_OUTLINES	1-30
CURSOR_SHARING	1-31
CURSOR_SPACE_FOR_TIME	1-32
DB_nK_CACHE_SIZE.....	1-32
DB_BLOCK_BUFFERS	1-33
DB_BLOCK_CHECKING.....	1-34
DB_BLOCK_CHECKSUM	1-34
DB_BLOCK_SIZE	1-35
DB_CACHE_ADVICE.....	1-35
DB_CACHE_SIZE	1-36
DB_CREATE_FILE_DEST	1-37
DB_CREATE_ONLINE_LOG_DEST_n	1-37
DB_DOMAIN	1-38
DB_FILE_MULTIBLOCK_READ_COUNT	1-38
DB_FILE_NAME_CONVERT	1-39
DB_FILES.....	1-40
DB_FLASHBACK_RETENTION_TARGET	1-41
DB_KEEP_CACHE_SIZE.....	1-41
DB_NAME	1-41
DB_RECOVERY_FILE_DEST	1-42
DB_RECOVERY_FILE_DEST_SIZE.....	1-43
DB_RECYCLE_CACHE_SIZE	1-43
DB_UNIQUE_NAME	1-43
DB_WRITER_PROCESSES.....	1-44
DBWR_IO_SLAVES.....	1-44
DDL_WAIT_FOR_LOCKS	1-45
DG_BROKER_CONFIG_FILEn	1-45
DG_BROKER_START.....	1-46

DISK_ASYNC_IO.....	1-46
DISPATCHERS.....	1-47
DISTRIBUTED_LOCK_TIMEOUT.....	1-50
DML_LOCKS.....	1-50
ENQUEUE_RESOURCES.....	1-51
EVENT.....	1-52
FAL_CLIENT.....	1-52
FAL_SERVER.....	1-52
FAST_START_IO_TARGET.....	1-53
FAST_START_MTTR_TARGET.....	1-54
FAST_START_PARALLEL_ROLLBACK.....	1-54
FILE_MAPPING.....	1-55
FILEIO_NETWORK_ADAPTERS.....	1-55
FILESYSTEMIO_OPTIONS.....	1-55
FIXED_DATE.....	1-56
GC_FILES_TO_LOCKS.....	1-56
GCS_SERVER_PROCESSES.....	1-57
GLOBAL_CONTEXT_POOL_SIZE.....	1-58
GLOBAL_NAMES.....	1-58
HASH_AREA_SIZE.....	1-58
HI_SHARED_MEMORY_ADDRESS.....	1-59
HS_AUTOREGISTER.....	1-59
IFILE.....	1-60
INSTANCE_GROUPS.....	1-60
INSTANCE_NAME.....	1-61
INSTANCE_NUMBER.....	1-61
INSTANCE_TYPE.....	1-62
JAVA_MAX_SESSIONSPACE_SIZE.....	1-63
JAVA_POOL_SIZE.....	1-63
JAVA_SOFT_SESSIONSPACE_LIMIT.....	1-64
JOB_QUEUE_PROCESSES.....	1-64
LARGE_POOL_SIZE.....	1-65
LDAP_DIRECTORY_ACCESS.....	1-65
LICENSE_MAX_SESSIONS.....	1-66
LICENSE_MAX_USERS.....	1-67
LICENSE_SESSIONS_WARNING.....	1-67
LOCAL_LISTENER.....	1-68
LOCK_NAME_SPACE.....	1-68
LOCK_SGA.....	1-69
LOG_ARCHIVE_CONFIG.....	1-69
LOG_ARCHIVE_DEST.....	1-70
LOG_ARCHIVE_DEST_n.....	1-71
LOG_ARCHIVE_DEST_STATE_n.....	1-72
LOG_ARCHIVE_DUPLEX_DEST.....	1-73
LOG_ARCHIVE_FORMAT.....	1-74
LOG_ARCHIVE_LOCAL_FIRST.....	1-75
LOG_ARCHIVE_MAX_PROCESSES.....	1-76

LOG_ARCHIVE_MIN_SUCCEED_DEST	1-76
LOG_ARCHIVE_TRACE	1-77
LOG_BUFFER	1-78
LOG_CHECKPOINT_INTERVAL	1-79
LOG_CHECKPOINT_TIMEOUT	1-80
LOG_CHECKPOINTS_TO_ALERT	1-81
LOG_FILE_NAME_CONVERT	1-81
LOGMNR_MAX_PERSISTENT_SESSIONS	1-82
MAX_COMMIT_PROPAGATION_DELAY	1-82
MAX_DISPATCHERS	1-83
MAX_DUMP_FILE_SIZE	1-84
MAX_ENABLED_ROLES	1-84
MAX_SHARED_SERVERS	1-85
NLS_CALENDAR	1-85
NLS_COMP	1-86
NLS_CURRENCY	1-86
NLS_DATE_FORMAT	1-87
NLS_DATE_LANGUAGE	1-87
NLS_DUAL_CURRENCY	1-88
NLS_ISO_CURRENCY	1-88
NLS_LANGUAGE	1-89
NLS_LENGTH_SEMANTICS	1-89
NLS_NCHAR_CONV_EXCP	1-90
NLS_NUMERIC_CHARACTERS	1-90
NLS_SORT	1-91
NLS_TERRITORY	1-91
NLS_TIMESTAMP_FORMAT	1-92
NLS_TIMESTAMP_TZ_FORMAT	1-92
O7_DICTIONARY_ACCESSIBILITY	1-93
OBJECT_CACHE_MAX_SIZE_PERCENT	1-93
OBJECT_CACHE_OPTIMAL_SIZE	1-94
OLAP_PAGE_POOL_SIZE	1-94
OPEN_CURSORS	1-95
OPEN_LINKS	1-95
OPEN_LINKS_PER_INSTANCE	1-96
OPTIMIZER_DYNAMIC_SAMPLING	1-96
OPTIMIZER_FEATURES_ENABLE	1-97
OPTIMIZER_INDEX_CACHING	1-99
OPTIMIZER_INDEX_COST_ADJ	1-99
OPTIMIZER_MODE	1-100
OS_AUTHENT_PREFIX	1-100
OS_ROLES	1-101
PARALLEL_ADAPTIVE_MULTI_USER	1-102
PARALLEL_AUTOMATIC_TUNING	1-102
PARALLEL_EXECUTION_MESSAGE_SIZE	1-103
PARALLEL_INSTANCE_GROUP	1-103
PARALLEL_MAX_SERVERS	1-104

PARALLEL_MIN_PERCENT	1-104
PARALLEL_MIN_SERVERS.....	1-105
PARALLEL_THREADS_PER_CPU	1-106
PGA_AGGREGATE_TARGET	1-106
PLSQL_CODE_TYPE.....	1-107
PLSQL_COMPILER_FLAGS	1-107
PLSQL_DEBUG	1-108
PLSQL_NATIVE_LIBRARY_DIR	1-109
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT.....	1-109
PLSQL_OPTIMIZE_LEVEL	1-109
PLSQL_V2_COMPATIBILITY.....	1-110
PLSQL_WARNINGS	1-111
PRE_PAGE_SGA	1-112
PROCESSES	1-112
QUERY_REWRITE_ENABLED	1-113
QUERY_REWRITE_INTEGRITY.....	1-114
RDBMS_SERVER_DN.....	1-114
READ_ONLY_OPEN_DELAYED.....	1-115
RECOVERY_PARALLELISM	1-115
REMOTE_ARCHIVE_ENABLE	1-116
REMOTE_DEPENDENCIES_MODE.....	1-116
REMOTE_LISTENER.....	1-117
REMOTE_LOGIN_PASSWORDFILE.....	1-117
REMOTE_OS_AUTHENT.....	1-118
REMOTE_OS_ROLES.....	1-118
REPLICATION_DEPENDENCY_TRACKING	1-119
RESOURCE_LIMIT	1-119
RESOURCE_MANAGER_PLAN.....	1-120
RESUMABLE_TIMEOUT.....	1-120
ROLLBACK_SEGMENTS.....	1-121
SERIAL_REUSE.....	1-122
SERVICE_NAMES	1-123
SESSION_CACHED_CURSORS.....	1-123
SESSION_MAX_OPEN_FILES	1-124
SESSIONS	1-124
SGA_MAX_SIZE	1-125
SGA_TARGET	1-125
SHADOW_CORE_DUMP.....	1-126
SHARED_MEMORY_ADDRESS	1-127
SHARED_POOL_RESERVED_SIZE.....	1-127
SHARED_POOL_SIZE.....	1-127
SHARED_SERVER_SESSIONS	1-128
SHARED_SERVERS	1-129
SKIP_UNUSABLE_INDEXES	1-129
SMTP_OUT_SERVER	1-130
SORT_AREA_RETAINED_SIZE.....	1-130
SORT_AREA_SIZE	1-131

SPFILE	1-132
SQL_TRACE.....	1-132
SQL92_SECURITY	1-133
SQLTUNE_CATEGORY.....	1-133
STANDBY_ARCHIVE_DEST	1-134
STANDBY_FILE_MANAGEMENT.....	1-134
STAR_TRANSFORMATION_ENABLED	1-135
STATISTICS_LEVEL	1-135
STREAMS_POOL_SIZE	1-137
TAPE_ASYNC_IO	1-137
THREAD	1-137
TIMED_OS_STATISTICS	1-138
TIMED_STATISTICS	1-139
TRACE_ENABLED	1-139
TRACEFILE_IDENTIFIER	1-140
TRANSACTIONS	1-141
TRANSACTIONS_PER_ROLLBACK_SEGMENT.....	1-141
UNDO_MANAGEMENT	1-142
UNDO_RETENTION	1-142
UNDO_TABLESPACE.....	1-143
USE_INDIRECT_DATA_BUFFERS.....	1-143
USER_DUMP_DEST	1-144
UTL_FILE_DIR	1-144
WORKAREA_SIZE_POLICY.....	1-145

Part II Static Data Dictionary Views

2 Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES

About Static Data Dictionary Views.....	2-1
Oracle Replication Views	2-2
Oracle Workspace Manager Views.....	2-3
Recovery Catalog Views	2-4
Static Data Dictionary View Descriptions.....	2-4
ALL_ALL_TABLES	2-4
ALL_APPLY	2-7
ALL_APPLY_CONFLICT_COLUMNS	2-8
ALL_APPLY_DML_HANDLERS	2-8
ALL_APPLY_ENQUEUE	2-9
ALL_APPLY_ERROR.....	2-9
ALL_APPLY_EXECUTE	2-10
ALL_APPLY_KEY_COLUMNS	2-10
ALL_APPLY_PARAMETERS	2-10
ALL_APPLY_PROGRESS	2-11
ALL_APPLY_TABLE_COLUMNS.....	2-11
ALL_ARGUMENTS.....	2-12
ALL_ASSOCIATIONS.....	2-13

ALL_AUDIT_POLICIES	2-14
ALL_AUDIT_POLICY_COLUMNS	2-14
ALL_AW_PS	2-15
ALL_AWS	2-15
ALL_BASE_TABLE_MVIEWS	2-16
ALL_CAPTURE	2-16
ALL_CAPTURE_EXTRA_ATTRIBUTES	2-18
ALL_CAPTURE_PARAMETERS	2-18
ALL_CAPTURE_PREPARED_DATABASE	2-19
ALL_CAPTURE_PREPARED_SCHEMAS	2-19
ALL_CAPTURE_PREPARED_TABLES	2-19
ALL_CATALOG	2-20
ALL_CLUSTER_HASH_EXPRESSIONS	2-20
ALL_CLUSTERS	2-20
ALL_COL_COMMENTS	2-21
ALL_COL_PRIVS	2-22
ALL_COL_PRIVS_MADE	2-22
ALL_COL_PRIVS_RECD	2-23
ALL_COLL_TYPES	2-23
ALL_CONS_COLUMNS	2-24
ALL_CONS_OBJ_COLUMNS	2-24
ALL_CONSTRAINTS	2-25
ALL_CONTEXT	2-26
ALL_DB_LINKS	2-26
ALL_DEF_AUDIT_OPTS	2-27
ALL_DEPENDENCIES	2-28
ALL_DIM_ATTRIBUTES	2-28
ALL_DIM_CHILD_OF	2-29
ALL_DIM_HIERARCHIES	2-29
ALL_DIM_JOIN_KEY	2-29
ALL_DIM_LEVEL_KEY	2-30
ALL_DIM_LEVELS	2-30
ALL_DIMENSIONS	2-31
ALL_DIRECTORIES	2-31
ALL_ERRORS	2-31
ALL_EVALUATION_CONTEXT_TABLES	2-32
ALL_EVALUATION_CONTEXT_VARS	2-33
ALL_EVALUATION_CONTEXTS	2-33
ALL_EXTERNAL_LOCATIONS	2-34
ALL_EXTERNAL_TABLES	2-34
ALL_IND_COLUMNS	2-35
ALL_IND_EXPRESSIONS	2-36
ALL_IND_PARTITIONS	2-36
ALL_IND_STATISTICS	2-37
ALL_IND_SUBPARTITIONS	2-39
ALL_INDEXES	2-40
ALL_INDEXTYPE_ARRAYTYPES	2-43

ALL_INDEXTYPE_COMMENTS	2-43
ALL_INDEXTYPE_OPERATORS	2-44
ALL_INDEXTYPES	2-44
ALL_INTERNAL_TRIGGERS	2-45
ALL_JAVA_ARGUMENTS	2-45
ALL_JAVA_CLASSES	2-46
ALL_JAVA_DERIVATIONS	2-47
ALL_JAVA_FIELDS	2-47
ALL_JAVA_IMPLEMENTES	2-48
ALL_JAVA_INNERS	2-49
ALL_JAVA_LAYOUTS	2-49
ALL_JAVA_METHODS	2-50
ALL_JAVA_NCOMPS	2-51
ALL_JAVA_RESOLVERS	2-52
ALL_JAVA_THROWS	2-52
ALL_JOBS	2-53
ALL_JOIN_IND_COLUMNS	2-54
ALL_LIBRARIES	2-54
ALL_LOB_PARTITIONS	2-54
ALL_LOB_SUBPARTITIONS	2-55
ALL_LOB_TEMPLATES	2-56
ALL_LOBS	2-57
ALL_LOG_GROUP_COLUMNS	2-58
ALL_LOG_GROUPS	2-58
ALL_METHOD_PARAMS	2-59
ALL_METHOD_RESULTS	2-59
ALL_MVIEW_AGGREGATES	2-60
ALL_MVIEW_ANALYSIS	2-61
ALL_MVIEW_COMMENTS	2-62
ALL_MVIEW_DETAIL_RELATIONS	2-62
ALL_MVIEW_JOINS	2-63
ALL_MVIEW_KEYS	2-63
ALL_MVIEW_LOGS	2-64
ALL_MVIEW_REFRESH_TIMES	2-65
ALL_MVIEWS	2-65
ALL_NESTED_TABLE_COLS	2-68
ALL_NESTED_TABLES	2-70
ALL_OBJ_COLATTRS	2-70
ALL_OBJECT_TABLES	2-71
ALL_OBJECTS	2-73
ALL_OPANCILLARY	2-74
ALL_OPARGUMENTS	2-74
ALL_OPBINDINGS	2-75
ALL_OPERATOR_COMMENTS	2-75
ALL_OPERATORS	2-76
ALL_OUTLINE_HINTS	2-76
ALL_OUTLINES	2-77

ALL_PART_COL_STATISTICS	2-77
ALL_PART_HISTOGRAMS	2-78
ALL_PART_INDEXES	2-79
ALL_PART_KEY_COLUMNS.....	2-80
ALL_PART_LOBS	2-81
ALL_PART_TABLES.....	2-82
ALL_PARTIAL_DROP_TABS.....	2-83
ALL_PENDING_CONV_TABLES	2-83
ALL_PLSQL_OBJECT_SETTINGS	2-84
ALL_POLICIES.....	2-85
ALL_POLICY_CONTEXTS	2-86
ALL_POLICY_GROUPS.....	2-86
ALL_PROCEDURES.....	2-87
ALL_PROPAGATION.....	2-87
ALL_PUBLISHED_COLUMNS.....	2-88
ALL_QUEUE_TABLES	2-88
ALL_QUEUES	2-89
ALL_REFRESH	2-90
ALL_REFRESH_CHILDREN	2-91
ALL_REFRESH_DEPENDENCIES.....	2-91
ALL_REFS.....	2-92
ALL_REGISTERED_MVIEWS.....	2-92
ALL_REGISTRY_BANNERS	2-93
ALL_REWRITE_EQUIVALENCES.....	2-93
ALL_RULE_SET_RULES	2-93
ALL_RULE_SETS	2-94
ALL_RULES.....	2-94
ALL_SCHEDULER_JOB_ARGS	2-95
ALL_SCHEDULER_JOB_CLASSES.....	2-96
ALL_SCHEDULER_JOB_LOG.....	2-96
ALL_SCHEDULER_JOB_RUN_DETAILS	2-97
ALL_SCHEDULER_JOBS.....	2-97
ALL_SCHEDULER_PROGRAM_ARGS.....	2-99
ALL_SCHEDULER_PROGRAMS	2-100
ALL_SCHEDULER_RUNNING_JOBS.....	2-101
ALL_SCHEDULER_SCHEDULES.....	2-101
ALL_SCHEDULER_WINDOW_DETAILS	2-102
ALL_SCHEDULER_WINDOW_GROUPS	2-102
ALL_SCHEDULER_WINDOW_LOG.....	2-103
ALL_SCHEDULER_WINDOWS.....	2-103
ALL_SCHEDULER_WINGROUP_MEMBERS.....	2-104
ALL_SEC_RELEVANT_COLS.....	2-104
ALL_SEQUENCES	2-105
ALL_SERVICES.....	2-105
ALL_SOURCE.....	2-106
ALL_SOURCE_TABLES	2-106
ALL_SQLJ_TYPE_ATTRS	2-107

ALL_SQLJ_TYPE_METHODS	2-107
ALL_SQLJ_TYPES	2-108
ALL_STORED_SETTINGS	2-109
ALL_STREAMS_GLOBAL_RULES	2-110
ALL_STREAMS_MESSAGE_CONSUMERS	2-110
ALL_STREAMS_MESSAGE_RULES	2-111
ALL_STREAMS_NEWLY_SUPPORTED	2-111
ALL_STREAMS_RULES	2-112
ALL_STREAMS_SCHEMA_RULES	2-113
ALL_STREAMS_TABLE_RULES	2-114
ALL_STREAMS_TRANSFORM_FUNCTION	2-115
ALL_STREAMS_UNSUPPORTED	2-116
ALL_SUBPART_COL_STATISTICS	2-116
ALL_SUBPART_HISTOGRAMS	2-117
ALL_SUBPART_KEY_COLUMNS	2-118
ALL_SUBPARTITION_TEMPLATES	2-118
ALL_SUBSCRIBED_COLUMNS	2-119
ALL_SUBSCRIBED_TABLES	2-119
ALL_SUBSCRIPTIONS	2-120
ALL_SUMDELTA	2-120
ALL_SYNONYMS	2-121
ALL_TAB_COL_STATISTICS	2-121
ALL_TAB_COLS	2-122
ALL_TAB_COLUMNS	2-124
ALL_TAB_COMMENTS	2-125
ALL_TAB_HISTOGRAMS	2-126
ALL_TAB_MODIFICATIONS	2-126
ALL_TAB_PARTITIONS	2-127
ALL_TAB_PRIVS	2-128
ALL_TAB_PRIVS_MADE	2-129
ALL_TAB_PRIVS_RECD	2-129
ALL_TAB_STATISTICS	2-130
ALL_TAB_SUBPARTITIONS	2-131
ALL_TABLES	2-132
ALL_TRIGGER_COLS	2-134
ALL_TRIGGERS	2-135
ALL_TYPE_ATTRS	2-135
ALL_TYPE_METHODS	2-136
ALL_TYPE_VERSIONS	2-137
ALL_TYPES	2-137
ALL_UNUSED_COL_TABS	2-138
ALL_UPDATABLE_COLUMNS	2-138
ALL_USERS	2-139
ALL_USTATS	2-139
ALL_VARRAYS	2-140
ALL_VIEWS	2-140
ALL_WARNING_SETTINGS	2-141

ALL_XML_SCHEMAS	2-142
ALL_XML_TAB_COLS.....	2-142
ALL_XML_TABLES	2-143
ALL_XML_VIEW_COLS.....	2-143
ALL_XML_VIEWS	2-144
AUDIT_ACTIONS.....	2-144
CATALOG.....	2-145
CAT.....	2-145
CHAINED_ROWS	2-145
CHANGE_SETS	2-145
CHANGE_SOURCES.....	2-146
CHANGE_TABLES	2-147
CLU	2-147
COL	2-147
COLS.....	2-147
DATABASE_PROPERTIES	2-147

3 Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS

DBA_2PC_NEIGHBORS	3-1
DBA_2PC_PENDING.....	3-1
DBA_ADVISOR_ACTIONS.....	3-2
DBA_ADVISOR_COMMANDS.....	3-3
DBA_ADVISOR_DEFINITIONS	3-3
DBA_ADVISOR_FINDINGS.....	3-4
DBA_ADVISOR_JOURNAL	3-4
DBA_ADVISOR_LOG.....	3-5
DBA_ADVISOR_OBJECT_TYPES.....	3-6
DBA_ADVISOR_OBJECTS	3-6
DBA_ADVISOR_PARAMETERS.....	3-7
DBA_ADVISOR_RATIONALE	3-8
DBA_ADVISOR_RECOMMENDATIONS	3-9
DBA_ADVISOR_SQLA_REC_SUM.....	3-10
DBA_ADVISOR_SQLA_WK_MAP	3-10
DBA_ADVISOR_SQLA_WK_STMTS	3-11
DBA_ADVISOR_SQLW_JOURNAL	3-12
DBA_ADVISOR_SQLW_PARAMETERS	3-12
DBA_ADVISOR_SQLW_STMTS.....	3-13
DBA_ADVISOR_SQLW_SUM.....	3-14
DBA_ADVISOR_SQLW_TABLES.....	3-15
DBA_ADVISOR_SQLW_TEMPLATES.....	3-15
DBA_ADVISOR_TASKS.....	3-16
DBA_ADVISOR_TEMPLATES.....	3-17
DBA_ADVISOR_USAGE.....	3-17
DBA_ALERT_HISTORY.....	3-18
DBA_ALL_TABLES	3-18
DBA_APPLICATION_ROLES.....	3-19
DBA_APPLY	3-19

DBA_APPLY_CONFLICT_COLUMNS	3-19
DBA_APPLY_DML_HANDLERS	3-19
DBA_APPLY_ENQUEUE	3-19
DBA_APPLY_ERROR	3-19
DBA_APPLY_EXECUTE	3-19
DBA_APPLY_INSTANTIATED_GLOBAL	3-20
DBA_APPLY_INSTANTIATED_OBJECTS	3-20
DBA_APPLY_INSTANTIATED_SCHEMAS	3-20
DBA_APPLY_KEY_COLUMNS	3-20
DBA_APPLY_PARAMETERS	3-21
DBA_APPLY_PROGRESS	3-21
DBA_APPLY_TABLE_COLUMNS	3-21
DBA_AQ_AGENT_PRIVS	3-21
DBA_AQ_AGENTS	3-21
DBA_ASSOCIATIONS	3-22
DBA_ATTRIBUTE_TRANSFORMATIONS	3-22
DBA_AUDIT_EXISTS	3-22
DBA_AUDIT_OBJECT	3-23
DBA_AUDIT_POLICIES	3-25
DBA_AUDIT_POLICY_COLUMNS	3-25
DBA_AUDIT_SESSION	3-25
DBA_AUDIT_STATEMENT	3-26
DBA_AUDIT_TRAIL	3-27
DBA_AW_PS	3-29
DBA_AWS	3-29
DBA_BASE_TABLE_MVIEWS	3-30
DBA_BLOCKERS	3-30
DBA_CAPTURE	3-30
DBA_CAPTURE_EXTRA_ATTRIBUTES	3-30
DBA_CAPTURE_PARAMETERS	3-30
DBA_CAPTURE_PREPARED_DATABASE	3-30
DBA_CAPTURE_PREPARED_SCHEMAS	3-30
DBA_CAPTURE_PREPARED_TABLES	3-31
DBA_CATALOG	3-31
DBA_CLU_COLUMNS	3-31
DBA_CLUSTER_HASH_EXPRESSIONS	3-31
DBA_CLUSTERS	3-31
DBA_COL_COMMENTS	3-31
DBA_COL_PRIVS	3-31
DBA_COLL_TYPES	3-32
DBA_COMMON_AUDIT_TRAIL	3-32
DBA_CONS_OBJ_COLUMNS	3-34
DBA_CONSTRAINTS	3-34
DBA_CONTEXT	3-34
DBA_DATA_FILES	3-34
DBA_DATAPUMP_JOBS	3-34
DBA_DATAPUMP_SESSIONS	3-35

DBA_DB_LINKS	3-35
DBA_DDL_LOCKS	3-35
DBA_DEPENDENCIES	3-36
DBA_DIM_ATTRIBUTES	3-36
DBA_DIM_CHILD_OF.....	3-36
DBA_DIM_HIERARCHIES	3-36
DBA_DIM_JOIN_KEY.....	3-36
DBA_DIM_LEVEL_KEY.....	3-37
DBA_DIM_LEVELS.....	3-37
DBA_DIMENSIONS	3-37
DBA_DIRECTORIES	3-37
DBA_DML_LOCKS.....	3-37
DBA_DMT_FREE_SPACE.....	3-38
DBA_DMT_USED_EXTENTS.....	3-38
DBA_ENABLED_AGGREGATIONS	3-38
DBA_ENABLED_TRACES.....	3-38
DBA_ERRORS	3-39
DBA_EVALUATION_CONTEXT_TABLES	3-39
DBA_EVALUATION_CONTEXT_VARS	3-39
DBA_EVALUATION_CONTEXTS	3-39
DBA_EXP_FILES	3-39
DBA_EXP_OBJECTS	3-40
DBA_EXP_VERSION	3-40
DBA_EXTENTS	3-40
DBA_EXTERNAL_LOCATIONS	3-41
DBA_EXTERNAL_TABLES	3-41
DBA_FEATURE_USAGE_STATISTICS	3-41
DBA_FGA_AUDIT_TRAIL.....	3-43
DBA_FREE_SPACE.....	3-44
DBA_FREE_SPACE_COALESCED	3-45
DBA_HIGH_WATER_MARK_STATISTICS	3-45
DBA_HIST_ACTIVE_SESS_HISTORY.....	3-46
DBA_HIST_BASELINE	3-47
DBA_HIST_BG_EVENT_SUMMARY.....	3-47
DBA_HIST_BUFFER_POOL_STAT.....	3-47
DBA_HIST_CR_BLOCK_SERVER.....	3-48
DBA_HIST_CURRENT_BLOCK_SERVER	3-49
DBA_HIST_DATABASE_INSTANCE	3-50
DBA_HIST_DATAFILE.....	3-50
DBA_HIST_DB_CACHE_ADVICE.....	3-50
DBA_HIST_DLM_MISC	3-51
DBA_HIST_ENQUEUE_STAT	3-51
DBA_HIST_EVENT_NAME.....	3-52
DBA_HIST_FILEMETRIC_HISTORY	3-52
DBA_HIST_FILESTATXS.....	3-52
DBA_HIST_INSTANCE_RECOVERY	3-53
DBA_HIST_JAVA_POOL_ADVICE.....	3-54

DBA_HIST_LATCH.....	3-55
DBA_HIST_LATCH_CHILDREN	3-56
DBA_HIST_LATCH_MISSES_SUMMARY	3-56
DBA_HIST_LATCH_NAME.....	3-57
DBA_HIST_LATCH_PARENT	3-57
DBA_HIST_LIBRARYCACHE	3-57
DBA_HIST_LOG.....	3-58
DBA_HIST_METRIC_NAME	3-59
DBA_HIST_MTTR_TARGET_ADVICE	3-59
DBA_HIST_OPTIMIZER_ENV	3-60
DBA_HIST_OSSTAT	3-60
DBA_HIST_OSSTAT_NAME.....	3-61
DBA_HIST_PARAMETER	3-61
DBA_HIST_PARAMETER_NAME	3-61
DBA_HIST_PGA_TARGET_ADVICE.....	3-61
DBA_HIST_PGASTAT.....	3-62
DBA_HIST_RESOURCE_LIMIT	3-63
DBA_HIST_ROLLSTAT.....	3-63
DBA_HIST_ROWCACHE_SUMMARY.....	3-64
DBA_HIST_SEG_STAT	3-64
DBA_HIST_SEG_STAT_OBJ.....	3-66
DBA_HIST_SERVICE_NAME	3-66
DBA_HIST_SERVICE_STAT	3-66
DBA_HIST_SERVICE_WAIT_CLASS.....	3-67
DBA_HIST_SESSMETRIC_HISTORY.....	3-67
DBA_HIST_SGA.....	3-67
DBA_HIST_SGASTAT.....	3-68
DBA_HIST_SHARED_POOL_ADVICE	3-68
DBA_HIST_SNAP_ERROR	3-69
DBA_HIST_SNAPSHOT.....	3-69
DBA_HIST_SQL_PLAN	3-69
DBA_HIST_SQL_SUMMARY.....	3-71
DBA_HIST_SQL_WORKAREA_HSTGRM.....	3-71
DBA_HIST_SQLBIND.....	3-72
DBA_HIST_SQLSTAT	3-72
DBA_HIST_SQLTEXT	3-75
DBA_HIST_STAT_NAME.....	3-75
DBA_HIST_SYS_TIME_MODEL	3-75
DBA_HIST_SYSMETRIC_HISTORY	3-75
DBA_HIST_SYSMETRIC_SUMMARY.....	3-76
DBA_HIST_SYSSTAT	3-76
DBA_HIST_SYSTEM_EVENT	3-77
DBA_HIST_TABLESPACE_STAT.....	3-77
DBA_HIST_TBSPC_SPACE_USAGE	3-77
DBA_HIST_TEMPFILE.....	3-78
DBA_HIST_TEMPSTATXS	3-78
DBA_HIST_THREAD	3-79

DBA_HIST_UNDOSTAT.....	3-79
DBA_HIST_WAITCLASSMET_HISTORY.....	3-80
DBA_HIST_WAITSTAT.....	3-81
DBA_HIST_WR_CONTROL.....	3-81
DBA_IND_COLUMNS.....	3-81
DBA_IND_EXPRESSIONS.....	3-81
DBA_IND_PARTITIONS.....	3-82
DBA_IND_STATISTICS.....	3-82
DBA_IND_SUBPARTITIONS.....	3-82
DBA_INDEXES.....	3-82
DBA_INDEXTYPE_ARRAYTYPES.....	3-82
DBA_INDEXTYPE_COMMENTS.....	3-82
DBA_INDEXTYPE_OPERATORS.....	3-82
DBA_INDEXTYPES.....	3-82
DBA_INTERNAL_TRIGGERS.....	3-83
DBA_JAVA_ARGUMENTS.....	3-83
DBA_JAVA_CLASSES.....	3-83
DBA_JAVA_DERIVATIONS.....	3-83
DBA_JAVA_FIELDS.....	3-83
DBA_JAVA_IMPLEMENTES.....	3-83
DBA_JAVA_INNERS.....	3-83
DBA_JAVA_LAYOUTS.....	3-83
DBA_JAVA_METHODS.....	3-84
DBA_JAVA_NCOMPS.....	3-84
DBA_JAVA_POLICY.....	3-84
DBA_JAVA_RESOLVERS.....	3-84
DBA_JAVA_THROWS.....	3-84
DBA_JOBS.....	3-85
DBA_JOBS_RUNNING.....	3-85
DBA_JOIN_IND_COLUMNS.....	3-85
DBA_KGLLOCK.....	3-85
DBA_LIBRARIES.....	3-85
DBA_LMT_FREE_SPACE.....	3-86
DBA_LMT_USED_EXTENTS.....	3-86
DBA_LOB_PARTITIONS.....	3-86
DBA_LOB_SUBPARTITIONS.....	3-86
DBA_LOB_TEMPLATES.....	3-86
DBA_LOBS.....	3-86
DBA_LOCK.....	3-87
DBA_LOCK_INTERNAL.....	3-87
DBA_LOCKS.....	3-87
DBA_LOG_GROUP_COLUMNS.....	3-87
DBA_LOG_GROUPS.....	3-87
DBA_LOGMNR_LOG.....	3-88
DBA_LOGMNR_PURGED_LOG.....	3-88
DBA_LOGMNR_SESSION.....	3-88
DBA_LOGSTDBY_EVENTS.....	3-89

DBA_LOGSTDBY_LOG	3-90
DBA_LOGSTDBY_NOT_UNIQUE.....	3-90
DBA_LOGSTDBY_PARAMETERS.....	3-91
DBA_LOGSTDBY_PROGRESS.....	3-91
DBA_LOGSTDBY_SKIP	3-92
DBA_LOGSTDBY_SKIP_TRANSACTION.....	3-92
DBA_LOGSTDBY_UNSUPPORTED.....	3-93
DBA_METHOD_PARAMS	3-93
DBA_METHOD_RESULTS.....	3-93
DBA_MVIEW_AGGREGATES	3-93
DBA_MVIEW_ANALYSIS.....	3-93
DBA_MVIEW_COMMENTS.....	3-94
DBA_MVIEW_DETAIL_RELATIONS.....	3-94
DBA_MVIEW_JOINS	3-94
DBA_MVIEW_KEYS	3-94
DBA_MVIEW_LOG_FILTER_COLS.....	3-94
DBA_MVIEW_LOGS	3-94
DBA_MVIEW_REFRESH_TIMES.....	3-95
DBA_MVIEWS	3-95
DBA_NESTED_TABLE_COLS	3-95
DBA_NESTED_TABLES.....	3-95
DBA_OBJ_AUDIT_OPTS.....	3-95
DBA_OBJ_COLATTRS	3-96
DBA_OBJECT_SIZE	3-96
DBA_OBJECT_TABLES.....	3-97
DBA_OBJECTS.....	3-97
DBA_OPANCILLARY	3-97
DBA_OPARGUMENTS	3-97
DBA_OPBINDINGS	3-97
DBA_OPERATOR_COMMENTS.....	3-97
DBA_OPERATORS.....	3-97
DBA_ORPHAN_KEY_TABLE	3-97
DBA_OUTLINE_HINTS.....	3-98
DBA_OUTLINES	3-98
DBA_OUTSTANDING_ALERTS	3-98
DBA_PART_COL_STATISTICS.....	3-99
DBA_PART_HISTOGRAMS	3-99
DBA_PART_INDEXES.....	3-99
DBA_PART_KEY_COLUMNS	3-99
DBA_PART_LOBS	3-99
DBA_PART_TABLES.....	3-99
DBA_PARTIAL_DROP_TABS.....	3-100
DBA_PENDING_CONV_TABLES.....	3-100
DBA_PENDING_TRANSACTIONS	3-100
DBA_PLSQL_OBJECT_SETTINGS	3-100
DBA_POLICIES.....	3-100
DBA_POLICY_CONTEXTS	3-100

DBA_POLICY_GROUPS.....	3-100
DBA_PRIV_AUDIT_OPTS.....	3-101
DBA_PROCEDURES.....	3-101
DBA_PROFILES.....	3-101
DBA_PROPAGATION.....	3-101
DBA_PROXIES.....	3-101
DBA_PUBLISHED_COLUMNS.....	3-102
DBA_QUEUE_SCHEDULES.....	3-102
DBA_QUEUE_TABLES.....	3-103
DBA_QUEUES.....	3-103
DBA_RCHILD.....	3-103
DBA_RECYCLEBIN.....	3-104
DBA_REDEFINITION_ERRORS.....	3-105
DBA_REDEFINITION_OBJECTS.....	3-105
DBA_REFRESH.....	3-106
DBA_REFRESH_CHILDREN.....	3-106
DBA_REFS.....	3-106
DBA_REGISTERED_ARCHIVED_LOG.....	3-106
DBA_REGISTERED_MVIEW_GROUPS.....	3-106
DBA_REGISTERED_MVIEWS.....	3-107
DBA_REGISTRY.....	3-107
DBA_REGISTRY_HIERARCHY.....	3-108
DBA_REPAIR_TABLE.....	3-108
DBA_RESUMABLE.....	3-109
DBA_REWRITE_EQUIVALENCES.....	3-110
DBA_RGROUP.....	3-110
DBA_ROLE_PRIVS.....	3-110
DBA_ROLES.....	3-110
DBA_ROLLBACK_SEGS.....	3-111
DBA_RSRC_CONSUMER_GROUP_PRIVS.....	3-111
DBA_RSRC_CONSUMER_GROUPS.....	3-112
DBA_RSRC_GROUP_MAPPINGS.....	3-112
DBA_RSRC_MANAGER_SYSTEM_PRIVS.....	3-112
DBA_RSRC_MAPPING_PRIORITY.....	3-113
DBA_RSRC_PLAN_DIRECTIVES.....	3-113
DBA_RSRC_PLANS.....	3-114
DBA_RULE_SET_RULES.....	3-115
DBA_RULE_SETS.....	3-115
DBA_RULES.....	3-115
DBA_SCHEDULER_JOB_ARGS.....	3-115
DBA_SCHEDULER_JOB_CLASSES.....	3-115
DBA_SCHEDULER_JOB_LOG.....	3-115
DBA_SCHEDULER_JOB_RUN_DETAILS.....	3-115
DBA_SCHEDULER_JOBS.....	3-115
DBA_SCHEDULER_PROGRAM_ARGS.....	3-116
DBA_SCHEDULER_PROGRAMS.....	3-116
DBA_SCHEDULER_RUNNING_JOBS.....	3-116

DBA_SCHEDULER_SCHEDULES.....	3-116
DBA_SCHEDULER_WINDOW_DETAILS	3-116
DBA_SCHEDULER_WINDOW_GROUPS	3-116
DBA_SCHEDULER_WINDOW_LOG	3-116
DBA_SCHEDULER_WINDOWS	3-117
DBA_SCHEDULER_WINDOWGROUP_MEMBERS.....	3-117
DBA_SEC_RELEVANT_COLS	3-117
DBA_SEGMENTS.....	3-117
DBA_SEQUENCES	3-118
DBA_SERVER_REGISTRY	3-118
DBA_SERVICES	3-118
DBA_SOURCE.....	3-119
DBA_SOURCE_TABLES	3-119
DBA_SQL_PROFILES.....	3-119
DBA_SQLJ_TYPE_ATTRS	3-119
DBA_SQLJ_TYPE_METHODS	3-119
DBA_SQLJ_TYPES	3-120
DBA_SQLSET	3-120
DBA_SQLSET_BINDS.....	3-120
DBA_SQLSET_REFERENCES.....	3-120
DBA_SQLSET_STATEMENTS.....	3-121
DBA_SQLTUNE_BINDS	3-122
DBA_SQLTUNE_PLANS.....	3-122
DBA_SQLTUNE_RATIONALE_PLAN	3-124
DBA_SQLTUNE_STATISTICS.....	3-124
DBA_STMT_AUDIT_OPTS	3-125
DBA_STORED_SETTINGS.....	3-125
DBA_STREAMS_ADMINISTRATOR	3-125
DBA_STREAMS_GLOBAL_RULES	3-126
DBA_STREAMS_MESSAGE_CONSUMERS.....	3-126
DBA_STREAMS_MESSAGE_RULES	3-126
DBA_STREAMS_NEWLY_SUPPORTED	3-126
DBA_STREAMS_RULES	3-126
DBA_STREAMS_SCHEMA_RULES	3-126
DBA_STREAMS_TABLE_RULES	3-126
DBA_STREAMS_TRANSFORM_FUNCTION	3-127
DBA_STREAMS_UNSUPPORTED	3-127
DBA_SUBPART_COL_STATISTICS.....	3-127
DBA_SUBPART_HISTOGRAMS.....	3-127
DBA_SUBPART_KEY_COLUMNS	3-127
DBA_SUBPARTITION_TEMPLATES	3-127
DBA_SUBSCRIBED_COLUMNS.....	3-127
DBA_SUBSCRIBED_TABLES	3-128
DBA_SUBSCRIPTIONS	3-128
DBA_SYNONYMS.....	3-128
DBA_SYS_PRIVS.....	3-128
DBA_TAB_COL_STATISTICS	3-128

DBA_TAB_COLS.....	3-128
DBA_TAB_COLUMNS.....	3-129
DBA_TAB_COMMENTS.....	3-129
DBA_TAB_HISTOGRAMS.....	3-129
DBA_TAB_MODIFICATIONS.....	3-129
DBA_TAB_PARTITIONS.....	3-129
DBA_TAB_PRIVS.....	3-129
DBA_TAB_STATISTICS.....	3-130
DBA_TAB_SUBPARTITIONS.....	3-130
DBA_TABLES.....	3-130
DBA_TABLESPACE_GROUPS.....	3-130
DBA_TABLESPACES.....	3-130
DBA_TEMP_FILES.....	3-132
DBA_THRESHOLDS.....	3-132
DBA_TRANSFORMATIONS.....	3-133
DBA_TRIGGER_COLS.....	3-133
DBA_TRIGGERS.....	3-133
DBA_TS_QUOTAS.....	3-133
DBA_TUNE_MVIEW.....	3-134
DBA_TYPE_ATTRS.....	3-134
DBA_TYPE_METHODS.....	3-134
DBA_TYPE_VERSIONS.....	3-134
DBA_TYPES.....	3-135
DBA_UNDO_EXTENTS.....	3-135
DBA_UNUSED_COL_TABS.....	3-135
DBA_UPDATABLE_COLUMNS.....	3-135
DBA_USERS.....	3-135
DBA_USTATS.....	3-136
DBA_VARRAYS.....	3-136
DBA_VIEWS.....	3-136
DBA_WAITERS.....	3-137
DBA_WARNING_SETTINGS.....	3-137
DBA_XML_SCHEMAS.....	3-137
DBA_XML_TAB_COLS.....	3-137
DBA_XML_TABLES.....	3-137
DBA_XML_VIEW_COLS.....	3-137
FLASHBACK_TRANSACTION_QUERY.....	3-138
GLOBAL_NAME.....	3-138
HS_ALL_CAPS.....	3-138
HS_ALL_DD.....	3-138
DBA_XML_VIEWS.....	3-139
DBMS_ALERT_INFO.....	3-139
DBMS_LOCK_ALLOCATED.....	3-139
DEPTREE.....	3-139
DICT.....	3-140
DICT_COLUMNS.....	3-140
DICTIONARY.....	3-140

ERROR_SIZE	3-140
EXCEPTIONS	3-140
HS_ALL_INITS	3-140
HS_BASE_CAPS	3-141
HS_BASE_DD	3-141
HS_CLASS_CAPS	3-141
HS_CLASS_DD	3-141
HS_CLASS_INIT	3-142
HS_EXTERNAL_OBJECT_PRIVILEGES	3-142
HS_EXTERNAL_OBJECTS	3-142
HS_EXTERNAL_USER_PRIVILEGES	3-143
HS_FDS_CLASS	3-143
HS_FDS_INST	3-143
HS_INST_CAPS	3-143
HS_INST_DD	3-144
HS_INST_INIT	3-144
IDEPTREE	3-145
IND	3-145
INDEX_HISTOGRAM	3-145
INDEX_STATS	3-145
MAP_OBJECT	3-146
NLS_DATABASE_PARAMETERS	3-147
NLS_INSTANCE_PARAMETERS	3-147
NLS_SESSION_PARAMETERS	3-147
OBJ	3-147
PLAN_TABLE	3-147
PLUGGABLE_SET_CHECK	3-149
PRODUCT_COMPONENT_VERSION	3-149
PROXY_USERS	3-150
PSTUBTBL	3-150
PUBLIC_DEPENDENCY	3-150
PUBLICSYN	3-150
QUEUE_PRIVILEGES	3-151
RECYCLEBIN	3-151
RESOURCE_COST	3-151
RESOURCE_MAP	3-151
ROLE_ROLE_PRIVS	3-151
ROLE_SYS_PRIVS	3-152
ROLE_TAB_PRIVS	3-152
SEQ	3-152
SESSION_CONTEXT	3-152
SESSION_PRIVS	3-152
SESSION_ROLES	3-153
SOURCE_SIZE	3-153
STMT_AUDIT_OPTION_MAP	3-153
SYN	3-153
SYNONYMS	3-153

SYSCATALOG	3-153
SYSFILES	3-153
SYSSEGOBJ.....	3-154
SYSTEM_PRIVILEGE_MAP	3-154
SYS_OBJECTS	3-154
TAB.....	3-154
TABLE_PRIVILEGE_MAP	3-154
TABS	3-154
TABQUOTAS	3-155
TRUSTED_SERVERS	3-155
TS_PITR_CHECK.....	3-155
TS_PITR_OBJECTS_TO_BE_DROPPED.....	3-156
UNI_PLUGGABLE_SET_CHECK	3-156
USER_ADVISOR_ACTIONS.....	3-157
USER_ADVISOR_FINDINGS	3-157
USER_ADVISOR_JOURNAL	3-157
USER_ADVISOR_LOG	3-157
USER_ADVISOR_OBJECTS	3-157
USER_ADVISOR_PARAMETERS	3-157
USER_ADVISOR_RATIONALE.....	3-157
USER_ADVISOR_RECOMMENDATIONS.....	3-158
USER_ADVISOR_SQLA_REC_SUM	3-158
USER_ADVISOR_SQLA_WK_MAP	3-158
USER_ADVISOR_SQLA_WK_STMTS.....	3-158
USER_ADVISOR_SQLW_JOURNAL.....	3-158
USER_ADVISOR_SQLW_PARAMETERS	3-158
USER_ADVISOR_SQLW_STMTS	3-158
USER_ADVISOR_SQLW_SUM.....	3-159
USER_ADVISOR_SQLW_TABLES	3-159
USER_ADVISOR_SQLW_TEMPLATES	3-159
USER_ADVISOR_TASKS.....	3-159
USER_ADVISOR_TEMPLATES.....	3-159
USER_ALL_TABLES.....	3-159
USER_AQ_AGENT_PRIVS	3-159
USER_ARGUMENTS.....	3-160
USER_ASSOCIATIONS	3-160
USER_ATTRIBUTE_TRANSFORMATIONS.....	3-160
USER_AUDIT_OBJECT.....	3-160
USER_AUDIT_POLICIES	3-160
USER_AUDIT_POLICY_COLUMNS	3-160
USER_AUDIT_SESSION	3-160
USER_AUDIT_STATEMENT	3-161
USER_AUDIT_TRAIL.....	3-161
USER_AW_PS	3-161
USER_AWS.....	3-161
USER_BASE_TABLE_MVIEWS.....	3-161
USER_CATALOG.....	3-161

USER_CLU_COLUMNS	3-161
USER_CLUSTER_HASH_EXPRESSIONS	3-161
USER_CLUSTERS	3-161
USER_COL_COMMENTS	3-162
USER_COL_PRIVS	3-162
USER_COL_PRIVS_MADE	3-162
USER_COL_PRIVS_REC'D	3-162
USER_COLL_TYPES	3-162
USER_CONS_COLUMNS	3-162
USER_CONS_OBJ_COLUMNS	3-162
USER_CONSTRAINTS	3-163
USER_DATAPUMP_JOBS	3-163
USER_DB_LINKS	3-163
USER_DEPENDENCIES	3-163
USER_DIM_ATTRIBUTES	3-163
USER_DIM_CHILD_OF	3-163
USER_DIM_HIERARCHIES	3-163
USER_DIM_JOIN_KEY	3-163
USER_DIM_LEVEL_KEY	3-164
USER_DIM_LEVELS	3-164
USER_DIMENSIONS	3-164
USER_ERRORS	3-164
USER_EVALUATION_CONTEXT_TABLES	3-164
USER_EVALUATION_CONTEXT_VARS	3-164
USER_EVALUATION_CONTEXTS	3-164
USER_EXTENTS	3-164
USER_EXTERNAL_LOCATIONS	3-165
USER_EXTERNAL_TABLES	3-165
USER_FREE_SPACE	3-165
USER_IND_COLUMNS	3-165
USER_IND_EXPRESSIONS	3-165
USER_IND_PARTITIONS	3-165
USER_IND_STATISTICS	3-165
USER_IND_SUBPARTITIONS	3-165
USER_INDEXES	3-166
USER_INDEXTYPE_ARRAYTYPES	3-166
USER_INDEXTYPE_COMMENTS	3-166
USER_INDEXTYPE_OPERATORS	3-166
USER_INDEXTYPES	3-166
USER_INTERNAL_TRIGGERS	3-166
USER_JAVA_ARGUMENTS	3-166
USER_JAVA_CLASSES	3-167
USER_JAVA_DERIVATIONS	3-167
USER_JAVA_FIELDS	3-167
USER_JAVA_IMPLEMENTES	3-167
USER_JAVA_INNERS	3-167
USER_JAVA_LAYOUTS	3-167

USER_JAVA_METHODS.....	3-167
USER_JAVA_NCOMPS.....	3-168
USER_JAVA_POLICY.....	3-168
USER_JAVA_RESOLVERS	3-168
USER_JAVA_THROWS	3-168
USER_JOBS.....	3-168
USER_JOIN_IND_COLUMNS	3-168
USER_LIBRARIES	3-168
USER_LOB_PARTITIONS.....	3-168
USER_LOB_SUBPARTITIONS.....	3-169
USER_LOB_TEMPLATES	3-169
USER_LOBS	3-169
USER_LOG_GROUP_COLUMNS	3-169
USER_LOG_GROUPS	3-169
USER_METHOD_PARAMS	3-169
USER_METHOD_RESULTS.....	3-169
USER_MVIEW_AGGREGATES	3-169
USER_MVIEW_ANALYSIS	3-170
USER_MVIEW_COMMENTS.....	3-170
USER_MVIEW_DETAIL_RELATIONS.....	3-170
USER_MVIEW_JOINS.....	3-170
USER_MVIEW_KEYS	3-170
USER_MVIEW_LOGS	3-170
USER_MVIEW_REFRESH_TIMES	3-170
USER_MVIEWS	3-171
USER_NESTED_TABLE_COLS	3-171
USER_NESTED_TABLES.....	3-171
USER_OBJ_AUDIT_OPTS.....	3-171
USER_OBJ_COLATTRS	3-171
USER_OBJECT_SIZE	3-171
USER_OBJECT_TABLES	3-171
USER_OBJECTS.....	3-171
USER_OPANCILLARY.....	3-172
USER_OPARGUMENTS	3-172
USER_OPBINDINGS.....	3-172
USER_OPERATOR_COMMENTS	3-172
USER_OPERATORS.....	3-172
USER_OUTLINE_HINTS.....	3-172
USER_OUTLINES.....	3-172
USER_PART_COL_STATISTICS	3-172
USER_PART_HISTOGRAMS	3-173
USER_PART_INDEXES	3-173
USER_PART_KEY_COLUMNS.....	3-173
USER_PART_LOBS.....	3-173
USER_PART_TABLES	3-173
USER_PARTIAL_DROP_TABS.....	3-173
USER_PASSWORD_LIMITS.....	3-173

USER_PENDING_CONV_TABLES	3-174
USER_PLSQL_OBJECT_SETTINGS	3-174
USER_POLICIES	3-174
USER_POLICY_CONTEXTS	3-174
USER_POLICY_GROUPS	3-174
USER_PROCEDURES	3-174
USER_PROXIES	3-174
USER_PUBLISHED_COLUMNS	3-175
USER_QUEUE_SCHEDULES	3-175
USER_QUEUE_TABLES	3-175
USER_QUEUES	3-175
USER_RECYCLEBIN	3-175
USER_REFRESH	3-175
USER_REFRESH_CHILDREN	3-175
USER_REFS	3-175
USER_REGISTERED_MVIEWS	3-176
USER_REGISTRY	3-176
USER_RESOURCE_LIMITS	3-176
USER_RESUMABLE	3-176
USER_REWRITE_EQUIVALENCES	3-176
USER_ROLE_PRIVS	3-176
USER_RSRC_CONSUMER_GROUP_PRIVS	3-177
USER_RSRC_MANAGER_SYSTEM_PRIVS	3-177
USER_RULE_SET_RULES	3-177
USER_RULE_SETS	3-177
USER_RULES	3-177
USER_SCHEDULER_JOB_ARGS	3-177
USER_SCHEDULER_JOB_LOG	3-177
USER_SCHEDULER_JOB_RUN_DETAILS	3-178
USER_SCHEDULER_JOBS	3-178
USER_SCHEDULER_PROGRAM_ARGS	3-178
USER_SCHEDULER_PROGRAMS	3-178
USER_SCHEDULER_RUNNING_JOBS	3-178
USER_SCHEDULER_SCHEDULES	3-178
USER_SEC_RELEVANT_COLS	3-178
USER_SEGMENTS	3-179
USER_SEQUENCES	3-179
USER_SOURCE	3-179
USER_SOURCE_TABLES	3-179
USER_SQLJ_TYPE_ATTRS	3-179
USER_SQLJ_TYPE_METHODS	3-179
USER_SQLJ_TYPES	3-179
USER_SQLSET	3-179
USER_SQLSET_BINDS	3-180
USER_SQLSET_REFERENCES	3-180
USER_SQLSET_STATEMENTS	3-180
USER_SQLTUNE_BINDS	3-180

USER_SQLTUNE_PLANS.....	3-180
USER_SQLTUNE_RATIONALE_PLAN.....	3-180
USER_SQLTUNE_STATISTICS.....	3-180
USER_STORED_SETTINGS.....	3-181
USER_SUBPART_COL_STATISTICS.....	3-181
USER_SUBPART_HISTOGRAMS.....	3-181
USER_SUBPART_KEY_COLUMNS.....	3-181
USER_SUBPARTITION_TEMPLATES.....	3-181
USER_SUBSCRIBED_COLUMNS.....	3-181
USER_SUBSCRIBED_TABLES.....	3-181
USER_SUBSCRIPTIONS.....	3-182
USER_SYNONYMS.....	3-182
USER_SYS_PRIVS.....	3-182
USER_TAB_COL_STATISTICS.....	3-182
USER_TAB_COLS.....	3-182
USER_TAB_COLUMNS.....	3-182
USER_TAB_COMMENTS.....	3-182
USER_TAB_HISTOGRAMS.....	3-182
USER_TAB_MODIFICATIONS.....	3-183
USER_TAB_PARTITIONS.....	3-183
USER_TAB_PRIVS.....	3-183
USER_TAB_PRIVS_MADE.....	3-183
USER_TAB_PRIVS_RECD.....	3-183
USER_TAB_STATISTICS.....	3-183
USER_TAB_SUBPARTITIONS.....	3-184
USER_TABLES.....	3-184
USER_TABLESPACES.....	3-184
USER_TRANSFORMATIONS.....	3-184
USER_TRIGGER_COLS.....	3-184
USER_TRIGGERS.....	3-184
USER_TS_QUOTAS.....	3-184
USER_TUNE_MVIEW.....	3-184
USER_TYPE_ATTRS.....	3-185
USER_TYPE_METHODS.....	3-185
USER_TYPE_VERSIONS.....	3-185
USER_TYPES.....	3-185
USER_UNUSED_COL_TABS.....	3-185
USER_UPDATABLE_COLUMNS.....	3-185
USER_USERS.....	3-185
USER_USTATS.....	3-186
USER_VARRAYS.....	3-186
USER_VIEWS.....	3-186
USER_WARNING_SETTINGS.....	3-186
USER_XML_SCHEMAS.....	3-186
USER_XML_TAB_COLS.....	3-186
USER_XML_TABLES.....	3-186
USER_XML_VIEW_COLS.....	3-186

Part III Dynamic Performance Views

4 Dynamic Performance (V\$) Views

About Dynamic Performance Views	4-1
V\$ Views.....	4-1
GV\$ Views.....	4-2
Dynamic Performance View Descriptions	4-2
V\$ACCESS.....	4-2
V\$ACTIVE_INSTANCES	4-2
V\$ACTIVE_SERVICES	4-3
V\$ACTIVE_SESS_POOL_MTH.....	4-3
V\$ACTIVE_SESSION_HISTORY	4-3
V\$ALERT_TYPES	4-5
V\$AQ	4-5
V\$ARCHIVE	4-5
V\$ARCHIVE_DEST.....	4-6
V\$ARCHIVE_DEST_STATUS.....	4-8
V\$ARCHIVE_GAP.....	4-9
V\$ARCHIVE_PROCESSES.....	4-10
V\$ARCHIVED_LOG	4-10
V\$ASM_ALIAS.....	4-12
V\$ASM_CLIENT	4-12
V\$ASM_DISK.....	4-13
V\$ASM_DISKGROUP.....	4-15
V\$ASM_FILE.....	4-15
V\$ASM_OPERATION	4-16
V\$ASM_TEMPLATE	4-17
V\$BACKUP	4-17
V\$BACKUP_ASYNC_IO	4-17
V\$BACKUP_CORRUPTION	4-18
V\$BACKUP_DATAFILE	4-19
V\$BACKUP_DEVICE.....	4-20
V\$BACKUP_FILES	4-20
V\$BACKUP_PIECE.....	4-22
V\$BACKUP_REDOLOG	4-23
V\$BACKUP_SET	4-24
V\$BACKUP_SPFIL.....	4-25
V\$BACKUP_SYNC_IO	4-25
V\$BGPROCESS.....	4-26
V\$BH.....	4-26
V\$BLOCK_CHANGE_TRACKING	4-27
V\$BUFFER_POOL.....	4-28
V\$BUFFER_POOL_STATISTICS	4-28
V\$BUFFERED_PUBLISHERS.....	4-29
V\$BUFFERED_QUEUES.....	4-29

V\$BUFFERED_SUBSCRIBERS	4-30
V\$CACHE	4-31
V\$CACHE_LOCK	4-32
V\$CACHE_TRANSFER	4-34
V\$CIRCUIT	4-34
V\$CLASS_CACHE_TRANSFER.....	4-35
V\$CLIENT_STATS.....	4-36
V\$CONTEXT	4-36
V\$CONTROLFILE	4-36
V\$CONTROLFILE_RECORD_SECTION.....	4-36
V\$COPY_CORRUPTION	4-37
V\$CR_BLOCK_SERVER.....	4-38
V\$CURRENT_BLOCK_SERVER	4-39
V\$DATABASE.....	4-39
V\$DATABASE_BLOCK_CORRUPTION.....	4-43
V\$DATABASE_INCARNATION	4-43
V\$DATAFILE.....	4-44
V\$DATAFILE_COPY	4-45
V\$DATAFILE_HEADER	4-46
V\$DATAGUARD_CONFIG	4-47
V\$DATAGUARD_STATUS.....	4-47
V\$DB_CACHE_ADVICE.....	4-48
V\$DB_OBJECT_CACHE.....	4-48
V\$DB_PIPES	4-49
V\$DBFILE	4-49
V\$DBLINK	4-49
V\$DELETED_OBJECT	4-50
V\$DISPATCHER	4-50
V\$DISPATCHER_CONFIG.....	4-51
V\$DISPATCHER_RATE.....	4-52
V\$ENABLEDPRIVS.....	4-55
V\$ENQUEUE_LOCK.....	4-55
V\$ENQUEUE_STAT	4-56
V\$EVENT_HISTOGRAM	4-56
V\$EVENT_NAME.....	4-57
V\$EVENTMETRIC	4-57
V\$EXECUTION	4-57
V\$FALSE_PING.....	4-58
V\$FAST_START_SERVERS	4-58
V\$FAST_START_TRANSACTIONS.....	4-59
V\$FILE_CACHE_TRANSFER	4-59
V\$FILE_HISTOGRAM	4-60
V\$FILEMETRIC.....	4-60
V\$FILEMETRIC_HISTORY	4-60
V\$FILESTAT	4-61
V\$FIXED_TABLE.....	4-61
V\$FIXED_VIEW_DEFINITION	4-62

V\$FLASHBACK_DATABASE_LOG	4-62
V\$FLASHBACK_DATABASE_STAT	4-62
V\$GC_ELEMENT	4-62
V\$GC_ELEMENTS_WITH_COLLISIONS	4-63
V\$GCSHVMMASTER_INFO	4-63
V\$GCSPFMASTER_INFO	4-63
V\$GES_BLOCKING_ENQUEUE	4-64
V\$GES_CONVERT_LOCAL	4-65
V\$GES_CONVERT_REMOTE	4-65
V\$GES_ENQUEUE	4-65
V\$GES_LATCH	4-66
V\$GES_RESOURCE	4-66
V\$GES_STATISTICS	4-67
V\$GLOBAL_BLOCKED_LOCKS	4-67
V\$GLOBAL_TRANSACTION	4-67
V\$HS_AGENT	4-68
V\$HS_PARAMETER	4-68
V\$HS_SESSION	4-68
V\$HVMMASTER_INFO	4-69
V\$INDEXED_FIXED_COLUMN	4-69
V\$INSTANCE	4-69
V\$INSTANCE_RECOVERY	4-70
V\$JAVA_LIBRARY_CACHE_MEMORY	4-71
V\$JAVA_POOL_ADVICE	4-72
V\$LATCH	4-72
V\$LATCH_CHILDREN	4-73
V\$LATCH_MISSES	4-74
V\$LATCH_PARENT	4-74
V\$LATCHHOLDER	4-74
V\$LATCHNAME	4-74
V\$LIBRARY_CACHE_MEMORY	4-75
V\$LIBRARYCACHE	4-75
V\$LICENSE	4-76
V\$LOADISTAT	4-76
V\$LOADPSTAT	4-76
V\$LOCK	4-76
V\$LOCK_ACTIVITY	4-78
V\$LOCKED_OBJECT	4-78
V\$LOG	4-78
V\$LOG_HISTORY	4-79
V\$LOGFILE	4-79
V\$LOGHIST	4-80
V\$LOGMNR_CONTENTS	4-80
V\$LOGMNR_DICTIONARY	4-82
V\$LOGMNR_LOGS	4-82
V\$LOGMNR_PARAMETERS	4-83
V\$LOGSTDBY	4-83

V\$LOGSTDBY_STATS	4-84
V\$MANAGED_STANDBY	4-85
V\$MAP_COMP_LIST	4-86
V\$MAP_ELEMENT	4-86
V\$MAP_EXT_ELEMENT	4-87
V\$MAP_FILE.....	4-87
V\$MAP_FILE_EXTENT	4-87
V\$MAP_FILE_IO_STACK.....	4-88
V\$MAP_LIBRARY	4-88
V\$MAP_SUBELEMENT	4-89
V\$METRICNAME	4-89
V\$MTTR_TARGET_ADVICE	4-89
V\$MVREFRESH.....	4-90
V\$MYSTAT	4-90

5 Dynamic Performance (V\$) Views: V\$NLS_PARAMETERS to V\$WAITSTAT

V\$NLS_PARAMETERS	5-1
V\$NLS_VALID_VALUES	5-1
V\$OBJECT_DEPENDENCY.....	5-1
V\$OBJECT_USAGE.....	5-2
V\$OBSOLETE_BACKUP_FILES.....	5-2
V\$OBSOLETE_PARAMETER	5-4
V\$OFFLINE_RANGE	5-4
V\$OPEN_CURSOR.....	5-4
V\$OPTION	5-5
V\$OSSTAT	5-5
V\$PARALLEL_DEGREE_LIMIT_MTH.....	5-6
V\$PARAMETER	5-6
V\$PARAMETER2	5-7
V\$PGA_TARGET_ADVICE.....	5-9
V\$PGA_TARGET_ADVICE_HISTOGRAM	5-9
V\$PGASTAT.....	5-10
V\$PQ_SESSTAT.....	5-11
V\$PQ_SLAVE	5-12
V\$PQ_SYSSTAT	5-12
V\$PQ_TQSTAT	5-13
V\$PROCESS	5-14
V\$PROPAGATION_RECEIVER	5-14
V\$PROPAGATION_SENDER.....	5-14
V\$PROXY_ARCHIVEDLOG	5-15
V\$PROXY_DATAFILE.....	5-16
V\$PWFILERS	5-17
V\$PX_PROCESS.....	5-17
V\$PX_PROCESS_SYSSTAT	5-18
V\$PX_SESSION.....	5-18
V\$PX_SESSTAT	5-19
V\$QUEUE	5-19

V\$QUEUEING_MTH	5-20
V\$RECOVER_FILE	5-20
V\$RECOVERY_FILE_DEST	5-20
V\$RECOVERY_FILE_STATUS	5-20
V\$RECOVERY_LOG	5-21
V\$RECOVERY_PROGRESS	5-21
V\$RECOVERY_STATUS	5-22
V\$REPLPROP	5-22
V\$REPLQUEUE	5-23
V\$REQDIST	5-23
V\$RESERVED_WORDS	5-24
V\$RESOURCE	5-24
V\$RESOURCE_LIMIT	5-24
V\$RMAN_CONFIGURATION	5-25
V\$RMAN_OUTPUT	5-25
V\$RMAN_STATUS	5-26
V\$ROLLNAME	5-26
V\$ROLLSTAT	5-26
V\$ROWCACHE	5-27
V\$ROWCACHE_PARENT	5-28
V\$ROWCACHE_SUBORDINATE	5-28
V\$RSRC_CONSUMER_GROUP	5-29
V\$RSRC_CONSUMER_GROUP_CPU_MTH	5-29
V\$RSRC_PLAN	5-29
V\$RSRC_PLAN_CPU_MTH	5-30
V\$RULE	5-30
V\$RULE_SET	5-31
V\$RULE_SET_AGGREGATE_STATS	5-31
V\$SEGMENT_STATISTICS	5-32
V\$SEGSTAT	5-33
V\$SEGSTAT_NAME	5-33
V\$SERV_MOD_ACT_STATS	5-33
V\$SERVICE_EVENT	5-33
V\$SERVICE_STATS	5-34
V\$SERVICE_WAIT_CLASS	5-34
V\$SERVICEMETRIC	5-35
V\$SERVICEMETRIC_HISTORY	5-35
V\$SERVICES	5-35
V\$SES_OPTIMIZER_ENV	5-35
V\$SESS_IO	5-36
V\$SESS_TIME_MODEL	5-36
V\$SESSION	5-37
V\$SESSION_CONNECT_INFO	5-42
V\$SESSION_CURSOR_CACHE	5-42
V\$SESSION_EVENT	5-43
V\$SESSION_LONGOPS	5-43
V\$SESSION_OBJECT_CACHE	5-44

V\$SESSION_WAIT	5-45
V\$SESSION_WAIT_CLASS	5-46
V\$SESSION_WAIT_HISTORY	5-46
V\$SESSMETRIC	5-47
V\$SESSTAT	5-47
V\$SGA	5-47
V\$SGA_CURRENT_RESIZE_OPS	5-48
V\$SGA_DYNAMIC_COMPONENTS	5-48
V\$SGA_DYNAMIC_FREE_MEMORY	5-49
V\$SGA_RESIZE_OPS	5-49
V\$SGAINFO	5-50
V\$SGASTAT	5-50
V\$SHARED_POOL_ADVICE	5-50
V\$SHARED_POOL_RESERVED	5-51
V\$SHARED_SERVER	5-51
V\$SHARED_SERVER_MONITOR	5-52
V\$SORT_SEGMENT	5-52
V\$SPPARAMETER	5-53
V\$SQL	5-53
V\$SQL_BIND_CAPTURE	5-55
V\$SQL_BIND_DATA	5-56
V\$SQL_BIND_METADATA	5-57
V\$SQL_CURSOR	5-57
V\$SQL_OPTIMIZER_ENV	5-58
V\$SQL_PLAN	5-58
V\$SQL_PLAN_STATISTICS	5-60
V\$SQL_PLAN_STATISTICS_ALL	5-61
V\$SQL_REDIRECTION	5-63
V\$SQL_SHARED_CURSOR	5-64
V\$SQL_SHARED_MEMORY	5-65
V\$SQL_WORKAREA	5-65
V\$SQL_WORKAREA_ACTIVE	5-66
V\$SQL_WORKAREA_HISTOGRAM	5-67
V\$SQLAREA	5-68
V\$SQLTEXT	5-69
V\$SQLTEXT_WITH_NEWLINES	5-70
V\$STANDBY_LOG	5-70
V\$STATISTICS_LEVEL	5-71
V\$STATNAME	5-71
V\$STREAMS_APPLY_COORDINATOR	5-72
V\$STREAMS_APPLY_READER	5-73
V\$STREAMS_APPLY_SERVER	5-74
V\$STREAMS_CAPTURE	5-76
V\$SUBCACHE	5-77
V\$SYS_OPTIMIZER_ENV	5-78
V\$SYS_TIME_MODEL	5-78
V\$SYSAUX_OCCUPANTS	5-78

V\$SYSMETRIC.....	5-79
V\$SYSMETRIC_HISTORY.....	5-79
V\$SYSMETRIC_SUMMARY.....	5-79
V\$SYSSTAT.....	5-80
V\$SYSTEM_CURSOR_CACHE.....	5-80
V\$SYSTEM_EVENT.....	5-80
V\$SYSTEM_PARAMETER.....	5-81
V\$SYSTEM_PARAMETER2.....	5-82
V\$SYSTEM_WAIT_CLASS.....	5-83
V\$TABLESPACE.....	5-83
V\$TEMP_CACHE_TRANSFER.....	5-84
V\$TEMP_EXTENT_MAP.....	5-84
V\$TEMP_EXTENT_POOL.....	5-84
V\$TEMP_HISTOGRAM.....	5-85
V\$TEMP_SPACE_HEADER.....	5-85
V\$TEMPFILE.....	5-85
V\$TEMPORARY_LOBS.....	5-86
V\$TEMPSEG_USAGE.....	5-86
V\$TEMPSTAT.....	5-87
V\$THREAD.....	5-87
V\$THRESHOLD_TYPES.....	5-88
V\$TIMER.....	5-88
V\$TIMEZONE_NAMES.....	5-88
V\$TRANSACTION.....	5-88
V\$TRANSACTION_ENQUEUE.....	5-89
V\$TRANSPORTABLE_PLATFORM.....	5-90
V\$TYPE_SIZE.....	5-90
V\$UNDOSTAT.....	5-91
V\$VERSION.....	5-92
V\$VPD_POLICY.....	5-92
V\$WAITCLASSMETRIC.....	5-92
V\$WAITCLASSMETRIC_HISTORY.....	5-93
V\$WAITSTAT.....	5-93

A Database Limits

Datatype Limits.....	A-1
Physical Database Limits.....	A-2
Logical Database Limits.....	A-3
Process and Runtime Limits.....	A-4

B SQL Scripts

Creating the Data Dictionary.....	B-1
Creating Additional Data Dictionary Structures.....	B-2
The "NO" Scripts.....	B-4
Upgrade and Downgrade Scripts.....	B-4
Java Scripts.....	B-5

C Oracle Wait Events

Displaying Wait Events	C-1
Wait Events and Parameters	C-1
Parameter Descriptions	C-10
Wait Event Descriptions.....	C-12
alter system set dispatchers	C-13
batched allocate scn lock request	C-13
BFILE check if exists	C-13
BFILE check if open	C-13
BFILE closure	C-13
BFILE get length	C-13
BFILE get name object	C-14
BFILE get path object	C-14
BFILE internal seek	C-14
BFILE open	C-14
BFILE read	C-14
buffer busy waits	C-15
buffer deadlock.....	C-15
buffer for checkpoint	C-15
buffer latch	C-16
buffer read retry	C-16
checkpoint completed.....	C-16
checkpoint range buffer not saved	C-16
control file parallel write	C-17
control file sequential read.....	C-17
control file single write	C-17
conversion file read.....	C-18
db file parallel read	C-18
db file parallel write.....	C-18
db file scattered read.....	C-18
db file sequential read	C-19
db file single write.....	C-19
DFS db file lock.....	C-19
DFS lock handle	C-20
direct path read.....	C-20
direct path write	C-20
dispatcher shutdown	C-21
dispatcher timer.....	C-21
duplicate cluster key	C-21
enqueue.....	C-21
file identify	C-22
file open	C-22
free buffer waits.....	C-22
free global transaction table entry	C-22
free process state object	C-22
global cache busy	C-22
global cache lock cleanup.....	C-23

global cache freelist wait	C-23
global cache null to s.....	C-23
global cache null to x	C-23
global cache open s	C-24
global cache open x	C-24
global cache s to x.....	C-24
inactive session	C-24
inactive transaction branch.....	C-25
index block split	C-25
instance recovery	C-25
instance state change	C-25
io done	C-26
kcl bg acks	C-26
latch activity	C-26
latch free	C-26
library cache load lock.....	C-27
library cache lock.....	C-27
library cache pin	C-28
lock manager wait for remote message	C-28
log buffer space.....	C-28
log file parallel write.....	C-28
log file sequential read	C-28
log file single write.....	C-29
log file switch (archiving needed)	C-29
log file switch (checkpoint incomplete)	C-29
log file switch (clearing log file)	C-29
log file switch completion	C-29
log file sync.....	C-30
log switch/archive	C-30
on-going SCN fetch to complete	C-30
pending global transaction(s).....	C-30
pipe get	C-30
pipe put.....	C-31
PL/SQL lock timer	C-31
pmon timer.....	C-31
process startup.....	C-31
PX dequeue wait.....	C-31
PX qref latch.....	C-32
PX server shutdown.....	C-32
PX signal server	C-32
queue messages	C-32
rdbms ipc message	C-33
rdbms ipc message block	C-33
rdbms ipc reply.....	C-33
redo wait.....	C-33
row cache lock	C-33
scginq AST call.....	C-34

single-task message.....	C-34
smon timer	C-34
SQL*Net break/reset to client	C-34
SQL*Net break/reset to dblink	C-34
SQL*Net message from client.....	C-35
SQL*Net message from dblink.....	C-35
SQL*Net message to client.....	C-35
SQL*Net message to dblink.....	C-35
SQL*Net more data from client.....	C-35
SQL*Net more data from dblink.....	C-36
SQL*Net more data to client.....	C-36
SQL*Net more data to dblink.....	C-36
switch logfile command	C-36
timer in sksawat	C-36
transaction	C-37
unbound tx	C-37
undo segment extension.....	C-37
undo segment recovery	C-37
undo segment tx slot.....	C-37
virtual circuit status	C-38
WMON goes to sleep.....	C-38
write complete waits.....	C-38
writes stopped by instance recovery or database suspension.....	C-38

D Oracle Enqueue Names

E Statistics Descriptions

Displaying Statistics	E-1
Statistics Descriptions	E-1

Index

Send Us Your Comments

Oracle Database Reference, 10g Release 1 (10.1)

Part No. B10755-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:

Oracle Corporation
Server Technologies Documentation
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This manual provides reference information about database initialization parameters, static data dictionary views, dynamic performance views, database limits, and SQL scripts that are part of the Oracle Database.

Oracle Database Reference contains information that describes the features and functionality of the Oracle Database (also known as the standard edition) and the Oracle Database Enterprise Edition products. The Oracle Database and the Oracle Database Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are optional. For example, to use application failover, you must have the Enterprise Edition with the Real Application Clusters option.

See Also: *Oracle Database New Features* for information about the differences between the Oracle Database and the Oracle Database Enterprise Edition and the features and options that are available to you.

This preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

Oracle Database Reference is intended for database administrators, system administrators, and database application developers.

To use this document, you need TO BE FAMILIAR WITH THE FOLLOWING:

- Oracle database management system (DBMS) concepts
- Your operating system environment

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive

technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

This document contains:

Part I, "Initialization Parameters"

This part describes the database initialization parameters that can be specified in a parameter file to start or configure an instance.

Chapter 1, "Initialization Parameters"

This chapter lists and describes all initialization parameters in alphabetical order.

Part II, "Static Data Dictionary Views"

This part describes the Oracle data dictionary tables and views, also known as static data dictionary views.

Chapter 2, "Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES"

Chapter 3, "Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS"

Chapter 2 and Chapter 3 list and describe all static data dictionary views in alphabetical order.

Part III, "Dynamic Performance Views"

This part describes the dynamic performance views, also known as the V\$ views.

Chapter 4, "Dynamic Performance (V\$) Views"

Chapter 5, "Dynamic Performance (V\$) Views: V\$NLS_PARAMETERS to V\$WAITSTAT"

Chapter 4 and Chapter 5 list and describe all dynamic performance views in alphabetical order.

Appendix A, "Database Limits"

This appendix lists the limits of values associated with database functions and objects.

Appendix B, "SQL Scripts"

This appendix describes the SQL scripts that are required for optimal operation of the Oracle Database.

Appendix C, "Oracle Wait Events"

This appendix describes some event names, wait times, and parameters for wait events displayed by the V\$SESSION_WAIT and V\$SYSTEM_EVENT views.

Appendix D, "Oracle Enqueue Names"

This appendix lists some enqueues used by the Oracle Database.

Appendix E, "Statistics Descriptions"

This appendix describes some statistics stored in the V\$SESSION_WAIT and V\$SYSSTAT dynamic performance views.

Related Documents

For more information, see these Oracle resources:

- *Oracle Database Concepts* for a comprehensive introduction to the concepts and terminology used in this manual
- *Oracle Database Administrator's Guide* for information about administering the Oracle Database
- *Oracle Database Upgrade Guide* for the procedures for upgrading a previous release of Oracle to the new release
- *Oracle Database SQL Reference* for information on Oracle's SQL commands and functions
- *Oracle Database Application Developer's Guide - Fundamentals* for information about developing database applications within the Oracle Database

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/membership/>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/documentation/>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>old_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```


The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt\ "system32 is the same as C:\WINNT\SYSTEM32
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\orann, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle Database Platform Guide for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

What's New in Oracle Database Reference?

This section describes new features of Oracle Database 10g release 1 (10.1) and provides pointers to additional information. For information on features that were new in earlier releases of the Oracle Database, refer to the documentation for the earlier release.

The following sections describe the new features in Oracle Database Reference:

- [Oracle Database 10g Release 1 \(10.1\) New Features](#)

Oracle Database 10g Release 1 (10.1) New Features

The following initialization parameters are new to this release:

ASM_DISKGROUPS
ASM_DISKSTRING
ASM_POWER_LIMIT
CREATE_STORED_OUTLINES
DB_FLASHBACK_RETENTION_TARGET
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_UNIQUE_NAME
DDL_WAIT_FOR_LOCKS
FILEIO_NETWORK_ADAPTERS
GCS_SERVER_PROCESSES
INSTANCE_TYPE
LDAP_DIRECTORY_ACCESS
LOG_ARCHIVE_CONFIG
LOG_ARCHIVE_LOCAL_FIRST
PLSQL_CODE_TYPE
PLSQL_DEBUG
PLSQL_OPTIMIZE_LEVEL
PLSQL_WARNINGS
RESUMABLE_TIMEOUT
SGA_TARGET
SKIP_UNUSABLE_INDEXES
SMTP_OUT_SERVER
SQLTUNE_CATEGORY
STREAMS_POOL_SIZE

See Also: [Chapter 1, "Initialization Parameters"](#) for more information about these initialization parameters

The following static data dictionary views are new to this release:

ALL_ Views	DBA_ Views	USER_ Views
	DBA_ADVISOR_ACTIONS	USER_ADVISOR_ACTIONS
	DBA_ADVISOR_COMMANDS	
	DBA_ADVISOR_DEFINITIONS	
	DBA_ADVISOR_FINDINGS	USER_ADVISOR_FINDINGS
	DBA_ADVISOR_JOURNAL	USER_ADVISOR_JOURNAL
	DBA_ADVISOR_LOG	USER_ADVISOR_LOG
	DBA_ADVISOR_OBJECT_TYPES	
	DBA_ADVISOR_OBJECTS	USER_ADVISOR_OBJECTS
	DBA_ADVISOR_PARAMETERS	USER_ADVISOR_PARAMETERS
	DBA_ADVISOR_RATIONALE	USER_ADVISOR_RATIONALE
	DBA_ADVISOR_RECOMMENDATIONS	USER_ADVISOR_RECOMMENDATIONS
	DBA_ADVISOR_SQLA_REC_SUM	USER_ADVISOR_SQLA_REC_SUM
	DBA_ADVISOR_SQLA_WK_MAP	USER_ADVISOR_SQLA_WK_MAP
	DBA_ADVISOR_SQLA_WK_STMTS	USER_ADVISOR_SQLA_WK_STMTS
	DBA_ADVISOR_SQLW_JOURNAL	USER_ADVISOR_SQLW_JOURNAL
	DBA_ADVISOR_SQLW_PARAMETERS	USER_ADVISOR_SQLW_PARAMETERS
	DBA_ADVISOR_SQLW_STMTS	USER_ADVISOR_SQLW_STMTS
	DBA_ADVISOR_SQLW_SUM	USER_ADVISOR_SQLW_SUM
	DBA_ADVISOR_SQLW_TABLES	USER_ADVISOR_SQLW_TABLES
	DBA_ADVISOR_SQLW_TEMPLATES	USER_ADVISOR_SQLW_TEMPLATES
	DBA_ADVISOR_TASKS	USER_ADVISOR_TASKS
	DBA_ADVISOR_TEMPLATES	USER_ADVISOR_TEMPLATES
	DBA_ADVISOR_USAGE	
	DBA_ALERT_HISTORY	
ALL_APPLY_ENQUEUE	DBA_APPLY_ENQUEUE	
ALL_APPLY_EXECUTE	DBA_APPLY_EXECUTE	
	DBA_APPLY_INSTANTIATED_GLOBAL	
	DBA_APPLY_INSTANTIATED_SCHEMAS	
ALL_APPLY_TABLE_COLUMNS	DBA_APPLY_TABLE_COLUMNS	
ALL_AUDIT_POLICY_COLUMNS	DBA_AUDIT_POLICY_COLUMNS	USER_AUDIT_POLICY_COLUMNS
ALL_AW_PS		
ALL_AWS		
ALL_CAPTURE_EXTRA_ATTRIBUTES	DBA_CAPTURE_EXTRA_ATTRIBUTES	
	DBA_COMMON_AUDIT_TRAIL	
	DBA_DATAPUMP_JOBS	USER_DATAPUMP_JOBS
	DBA_DATAPUMP_SESSIONS	
	DBA_ENABLED_AGGREGATIONS	
	DBA_ENABLED_TRACES	
	DBA_FEATURE_USAGE_STATISTICS	
	DBA_HIGH_WATER_MARK_STATISTICS	
	DBA_HIST_ACTIVE_SESS_HISTORY	

ALL_ Views

DBA_ Views**USER_ Views**

DBA_HIST_BASELINE
DBA_HIST_BG_EVENT_SUMMARY
DBA_HIST_BUFFER_POOL_STAT
DBA_HIST_CR_BLOCK_SERVER
DBA_HIST_CURRENT_BLOCK_SERVER
DBA_HIST_DATABASE_INSTANCE
DBA_HIST_DATAFILE
DBA_HIST_DB_CACHE_ADVICE
DBA_HIST_DLM_MISC
DBA_HIST_ENQUEUE_STAT
DBA_HIST_EVENT_NAME
DBA_HIST_FILEMETRIC_HISTORY
DBA_HIST_FILESTATXS
DBA_HIST_INSTANCE_RECOVERY
DBA_HIST_JAVA_POOL_ADVICE
DBA_HIST_LATCH
DBA_HIST_LATCH_CHILDREN
DBA_HIST_LATCH_MISSES_SUMMARY
DBA_HIST_LATCH_NAME
DBA_HIST_LATCH_PARENT
DBA_HIST_LIBRARYCACHE
DBA_HIST_LOG
DBA_HIST_METRIC_NAME
DBA_HIST_MTTR_TARGET_ADVICE
DBA_HIST_OPTIMIZER_ENV
DBA_HIST_OSSTAT
DBA_HIST_OSSTAT_NAME
DBA_HIST_PARAMETER
DBA_HIST_PARAMETER_NAME
DBA_HIST_PGA_TARGET_ADVICE
DBA_HIST_PGASTAT
DBA_HIST_RESOURCE_LIMIT
DBA_HIST_ROLLSTAT
DBA_HIST_ROWCACHE_SUMMARY
DBA_HIST_SEG_STAT
DBA_HIST_SEG_STAT_OBJ
DBA_HIST_SERVICE_NAME
DBA_HIST_SERVICE_STAT
DBA_HIST_SERVICE_WAIT_CLASS
DBA_HIST_SESSMETRIC_HISTORY
DBA_HIST_SGA
DBA_HIST_SGASTAT
DBA_HIST_SHARED_POOL_ADVICE

ALL_ Views	DBA_ Views	USER_ Views
	DBA_HIST_SNAP_ERROR	
	DBA_HIST_SNAPSHOT	
	DBA_HIST_SQL_PLAN	
	DBA_HIST_SQL_SUMMARY	
	DBA_HIST_SQL_WORKAREA_HSTGRM	
	DBA_HIST_SQLBIND	
	DBA_HIST_SQLSTAT	
	DBA_HIST_SQLTEXT	
	DBA_HIST_STAT_NAME	
	DBA_HIST_SYS_TIME_MODEL	
	DBA_HIST_SYSMETRIC_HISTORY	
	DBA_HIST_SYSMETRIC_SUMMARY	
	DBA_HIST_SYSSTAT	
	DBA_HIST_SYSTEM_EVENT	
	DBA_HIST_TABLESPACE_STAT	
	DBA_HIST_TBSPC_SPACE_USAGE	
	DBA_HIST_TEMPFILE	
	DBA_HIST_TEMPSTATXS	
	DBA_HIST_THREAD	
	DBA_HIST_UNDOSTAT	
	DBA_HIST_WAITCLASSMET_HISTORY	
	DBA_HIST_WAITSTAT	
	DBA_HIST_WR_CONTROL	
ALL_IND_STATISTICS	DBA_IND_STATISTICS	USER_IND_STATISTICS
ALL_INDEXTYPE_ARRAYTYPES	DBA_INDEXTYPE_ARRAYTYPES	USER_INDEXTYPE_ARRAYTYPES
	DBA_LOGMNR_LOG	
	DBA_LOGMNR_PURGED_LOG	
	DBA_LOGMNR_SESSION	
ALL_MVIEW_COMMENTS	DBA_MVIEW_COMMENTS	USER_MVIEW_COMMENTS
ALL_NESTED_TABLE_COLS	DBA_NESTED_TABLE_COLS	USER_NESTED_TABLE_COLS
	DBA_OUTSTANDING_ALERTS	
ALL_PLSQL_OBJECT_SETTINGS	DBA_PLSQL_OBJECT_SETTINGS	USER_PLSQL_OBJECT_SETTINGS
	DBA_RECYCLEBIN	USER_RECYCLEBIN
	DBA_REDEFINITION_ERRORS	
	DBA_REDEFINITION_OBJECTS	
	DBA_REGISTERED_ARCHIVED_LOG	
ALL_REWRITE_EQUIVALENCES	DBA_REWRITE_EQUIVALENCES	USER_REWRITE_EQUIVALENCES
	DBA_RSRC_GROUP_MAPPINGS	
	DBA_RSRC_MAPPING_PRIORITY	
ALL_SCHEDULER_JOB_ARGS	DBA_SCHEDULER_JOB_ARGS	USER_SCHEDULER_JOB_ARGS
ALL_SCHEDULER_JOB_CLASSES	DBA_SCHEDULER_JOB_CLASSES	
ALL_SCHEDULER_JOB_LOG	DBA_SCHEDULER_JOB_LOG	USER_SCHEDULER_JOB_LOG
ALL_SCHEDULER_JOB_RUN_DETAILS	DBA_SCHEDULER_JOB_RUN_DETAILS	USER_SCHEDULER_JOB_RUN_DETAILS

ALL_ Views	DBA_ Views	USER_ Views
ALL_SCHEDULER_JOBS	DBA_SCHEDULER_JOBS	USER_SCHEDULER_JOBS
ALL_SCHEDULER_PROGRAM_ARGS	DBA_SCHEDULER_PROGRAM_ARGS	USER_SCHEDULER_PROGRAM_ARGS
ALL_SCHEDULER_PROGRAMS	DBA_SCHEDULER_PROGRAMS	USER_SCHEDULER_PROGRAMS
ALL_SCHEDULER_RUNNING_JOBS	DBA_SCHEDULER_RUNNING_JOBS	USER_SCHEDULER_RUNNING_JOBS
ALL_SCHEDULER_SCHEDULES	DBA_SCHEDULER_SCHEDULES	USER_SCHEDULER_SCHEDULES
ALL_SCHEDULER_WINDOW_DETAILS	DBA_SCHEDULER_WINDOW_DETAILS	
ALL_SCHEDULER_WINDOW_GROUPS	DBA_SCHEDULER_WINDOW_GROUPS	
ALL_SCHEDULER_WINDOW_LOG	DBA_SCHEDULER_WINDOW_LOG	
ALL_SCHEDULER_WINDOWS	DBA_SCHEDULER_WINDOWS	
ALL_SCHEDULER_WINGROUP_MEMBERS	DBA_SCHEDULER_WINGROUP_MEMBERS	
ALL_SEC_RELEVANT_COLS	DBA_SEC_RELEVANT_COLS	USER_SEC_RELEVANT_COLS
	DBA_SERVER_REGISTRY	
ALL_SERVICES	DBA_SERVICES	
	DBA_SQL_PROFILES	
	DBA_SQLSET	USER_SQLSET
	DBA_SQLSET_BINDS	USER_SQLSET_BINDS
	DBA_SQLSET_REFERENCES	USER_SQLSET_REFERENCES
	DBA_SQLSET_STATEMENTS	USER_SQLSET_STATEMENTS
	DBA_SQLTUNE_BINDS	USER_SQLTUNE_BINDS
	DBA_SQLTUNE_PLANS	USER_SQLTUNE_PLANS
	DBA_SQLTUNE_RATIONALE_PLAN	USER_SQLTUNE_RATIONALE_PLAN
	DBA_SQLTUNE_STATISTICS	USER_SQLTUNE_STATISTICS
	DBA_STREAMS_ADMINISTRATOR	
ALL_STREAMS_MESSAGE_CONSUMERS	DBA_STREAMS_MESSAGE_CONSUMERS	
ALL_STREAMS_MESSAGE_RULES	DBA_STREAMS_MESSAGE_RULES	
ALL_STREAMS_NEWLY_SUPPORTED	DBA_STREAMS_NEWLY_SUPPORTED	
ALL_STREAMS_RULES	DBA_STREAMS_RULES	
ALL_STREAMS_TRANSFORM_FUNCTION	DBA_STREAMS_TRANSFORM_FUNCTION	
ALL_STREAMS_UNSUPPORTED	DBA_STREAMS_UNSUPPORTED	
ALL_TAB_STATISTICS	DBA_TAB_STATISTICS	USER_TAB_STATISTICS
	DBA_TABLESPACE_GROUPS	
	DBA_THRESHOLDS	
	DBA_TUNE_MVIEW	USER_TUNE_MVIEW
ALL_WARNING_SETTINGS	DBA_WARNING_SETTINGS	USER_WARNING_SETTINGS

See Also: [Chapter 2, "Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES"](#) and [Chapter 3, "Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS"](#) for more information about these static data dictionary views

The following dynamic performance views are new to this release:

GV\$ Views	V\$ Views
GV\$ACTIVE_SERVICES	V\$ACTIVE_SERVICES

GV\$ Views	V\$ Views
GV\$ACTIVE_SESSION_HISTORY	V\$ACTIVE_SESSION_HISTORY
GV\$ALERT_TYPES	V\$ALERT_TYPES
GV\$ASM_ALIAS	V\$ASM_ALIAS
GV\$ASM_CLIENT	V\$ASM_CLIENT
GV\$ASM_DISK	V\$ASM_DISK
GV\$ASM_DISKGROUP	V\$ASM_DISKGROUP
GV\$ASM_FILE	V\$ASM_FILE
GV\$ASM_OPERATION	V\$ASM_OPERATION
GV\$ASM_TEMPLATE	V\$ASM_TEMPLATE
	V\$BACKUP_FILES
	V\$BLOCK_CHANGE_TRACKING
GV\$BUFFERED_PUBLISHERS	V\$BUFFERED_PUBLISHERS
GV\$BUFFERED_QUEUES	V\$BUFFERED_QUEUES
GV\$BUFFERED_SUBSCRIBERS	V\$BUFFERED_SUBSCRIBERS
GV\$CLIENT_STATS	V\$CLIENT_STATS
GV\$CURRENT_BLOCK_SERVER	V\$CURRENT_BLOCK_SERVER
GV\$DATAGUARD_CONFIG	V\$DATAGUARD_CONFIG
GV\$DISPATCHER_CONFIG	V\$DISPATCHER_CONFIG
GV\$EVENT_HISTOGRAM	V\$EVENT_HISTOGRAM
GV\$EVENTMETRIC	V\$EVENTMETRIC
GV\$FILE_HISTOGRAM	V\$FILE_HISTOGRAM
GV\$FILEMETRIC	V\$FILEMETRIC
GV\$FILEMETRIC_HISTORY	V\$FILEMETRIC_HISTORY
GV\$FLASHBACK_DATABASE_LOG	V\$FLASHBACK_DATABASE_LOG
GV\$FLASHBACK_DATABASE_STAT	V\$FLASHBACK_DATABASE_STAT
GV\$JAVA_LIBRARY_CACHE_MEMORY	V\$JAVA_LIBRARY_CACHE_MEMORY
GV\$JAVA_POOL_ADVICE	V\$JAVA_POOL_ADVICE
GV\$METRICNAME	V\$METRICNAME
GV\$OSSTAT	V\$OSSTAT
GV\$PROPAGATION_RECEIVER	V\$PROPAGATION_RECEIVER
GV\$PROPAGATION_SENDER	V\$PROPAGATION_SENDER
	V\$RECOVERY_FILE_DEST
GV\$RMAN_OUTPUT	V\$RMAN_OUTPUT
	V\$RMAN_STATUS
GV\$RULE	V\$RULE
GV\$RULE_SET	V\$RULE_SET
GV\$RULE_SET_AGGREGATE_STATS	V\$RULE_SET_AGGREGATE_STATS
GV\$SERV_MOD_ACT_STATS	V\$SERV_MOD_ACT_STATS

GV\$ Views	V\$ Views
GV\$SERVICE_EVENT	V\$SERVICE_EVENT
GV\$SERVICE_STATS	V\$SERVICE_STATS
GV\$SERVICE_WAIT_CLASS	V\$SERVICE_WAIT_CLASS
GV\$SERVICEMETRIC	V\$SERVICEMETRIC
GV\$SERVICEMETRIC_HISTORY	V\$SERVICEMETRIC_HISTORY
GV\$SERVICES	V\$SERVICES
GV\$SES_OPTIMIZER_ENV	V\$SES_OPTIMIZER_ENV
GV\$SESS_TIME_MODEL	V\$SESS_TIME_MODEL
GV\$SESSION_WAIT_CLASS	V\$SESSION_WAIT_CLASS
GV\$SESSION_WAIT_HISTORY	V\$SESSION_WAIT_HISTORY
GV\$SESSMETRIC	V\$SESSMETRIC
GV\$SGAINFO	V\$SGAINFO
GV\$SQL_BIND_CAPTURE	V\$SQL_BIND_CAPTURE
GV\$SQL_OPTIMIZER_ENV	V\$SQL_OPTIMIZER_ENV
GV\$SYS_OPTIMIZER_ENV	V\$SYS_OPTIMIZER_ENV
GV\$SYS_TIME_MODEL	V\$SYS_TIME_MODEL
GV\$SYS_AUX_OCCUPANTS	V\$SYS_AUX_OCCUPANTS
GV\$SYSMETRIC	V\$SYSMETRIC
GV\$SYSMETRIC_HISTORY	V\$SYSMETRIC_HISTORY
GV\$SYSMETRIC_SUMMARY	V\$SYSMETRIC_SUMMARY
GV\$SYSTEM_WAIT_CLASS	V\$SYSTEM_WAIT_CLASS
GV\$TEMP_HISTOGRAM	V\$TEMP_HISTOGRAM
GV\$THRESHOLD_TYPES	V\$THRESHOLD_TYPES
	V\$TRANSPORTABLE_PLATFORM
GV\$WAITCLASSMETRIC	V\$WAITCLASSMETRIC
GV\$WAITCLASSMETRIC_HISTORY	V\$WAITCLASSMETRIC_HISTORY

See Also: [Chapter 4, "Dynamic Performance \(V\\$\) Views"](#) and [Chapter 5, "Dynamic Performance \(V\\$\) Views: V\\$NLS_PARAMETERS to V\\$WAITSTAT"](#) for more information about these dynamic performance views

The following static data dictionary views have new columns in this release:

Static Data Dictionary View	New Columns
ALL_ALL_TABLES	DROPPED

Static Data Dictionary View	New Columns
ALL_APPLY	PRECOMMIT_HANDLER MAX_APPLIED_MESSAGE_NUMBER NEGATIVE_RULE_SET_NAME NEGATIVE_RULE_SET_OWNER STATUS_CHANGE_TIME ERROR_NUMBER ERROR_MESSAGE
ALL_APPLY_DML_HANDLERS	APPLY_NAME
ALL_APPLY_PROGRESS	SOURCE_DATABASE
ALL_AUDIT_POLICIES	SEL INS UPD DEL AUDIT_TRAIL POLICY_COLUMN_OPTIONS
ALL_CAPTURE	CAPTURE_USER CAPTURED_SCN APPLIED_SCN USE_DATABASE_LINK FIRST_SCN SOURCE_DATABASE SOURCE_DBID SOURCE_RESETLOGS_SCN SOURCE_RESETLOGS_TIME LOGMINER_ID NEGATIVE_RULE_SET_NAME NEGATIVE_RULE_SET_OWNER MAX_CHECKPOINT_SCN REQUIRED_CHECKPOINT_SCN LOGFILE_ASSIGNMENT STATUS_CHANGE_TIME ERROR_NUMBER ERROR_MESSAGE VERSION CAPTURE_TYPE
ALL_DEF_AUDIT_OPTS	FBK
ALL_DIM_ATTRIBUTES	ATTRIBUTE_NAME
ALL_ERRORS	ATTRIBUTE MESSAGE_NUMBER
ALL_EXTERNAL_TABLES	PROPERTY

Static Data Dictionary View	New Columns
ALL_INDEXES	IOT_REDUNDANT_PKEY_ELIM DROPPED
ALL_INDEXTYPES	ARRAY_DML
ALL_LOBS	TABLESPACE_NAME FORMAT PARTITIONED
ALL_LOG_GROUP_COLUMNS	LOGGING_PROPERTY
ALL_LOG_GROUPS	LOG_GROUP_TYPE GENERATED
ALL_MVIEWS	UNKNOWN_TRUSTED_FD STALE_SINCE
ALL_OBJECT_TABLES	DROPPED
ALL_OUTLINES	COMPATIBLE ENABLED FORMAT
ALL_PART_COL_STATISTICS	HISTOGRAM
ALL_POLICIES	IDX POLICY_TYPE LONG_PREDICATE
ALL_PROPAGATION	NEGATIVE_RULE_SET_OWNER NEGATIVE_RULE_SET_NAME
ALL_SUBPART_COL_STATISTICS	HISTOGRAM
ALL_SUBSCRIBED_COLUMNS	SUBSCRIPTION_NAME
ALL_SUBSCRIBED_TABLES	SUBSCRIPTION_NAME
ALL_SUBSCRIPTIONS	SUBSCRIPTION_NAME
ALL_TAB_COL_STATISTICS	HISTOGRAM
ALL_TAB_COLS	HISTOGRAM QUALIFIED_COL_NAME
ALL_TAB_COLUMNS	HISTOGRAM
ALL_TAB_MODIFICATIONS	DROP_SEGMENTS
ALL_TABLES	DROPPED
DBA_ALL_TABLES	DROPPED
DBA_APPLY	PRECOMMIT_HANDLER MAX_APPLIED_MESSAGE_NUMBER NEGATIVE_RULE_SET_NAME NEGATIVE_RULE_SET_OWNER STATUS_CHANGE_TIME ERROR_NUMBER ERROR_MESSAGE
DBA_APPLY_DML_HANDLERS	APPLY_NAME

Static Data Dictionary View	New Columns
DBA_APPLY_PROGRESS	SOURCE_DATABASE
DBA_AUDIT_EXISTS	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
	TRANSACTIONID
	SCN
	SQL_BIND
	SQL_TEXT
DBA_AUDIT_OBJECT	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
	TRANSACTIONID
	SCN
	SQL_BIND
	SQL_TEXT
DBA_AUDIT_POLICIES	SEL
	INS
	UPD
	DEL
	AUDIT_TRAIL
	POLICY_COLUMN_OPTIONS
DBA_AUDIT_SESSION	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
DBA_AUDIT_STATEMENT	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
	TRANSACTIONID
	SCN
	SQL_BIND
	SQL_TEXT

Static Data Dictionary View

New Columns

DBA_AUDIT_TRAIL	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
	TRANSACTIONID
	SCN
	SQL_BIND
	SQL_TEXT
DBA_AW_PS	MAXPAGES
DBA_CAPTURE	CAPTURE_USER
	USE_DATABASE_LINK
	FIRST_SCN
	SOURCE_DATABASE
	SOURCE_DBID
	SOURCE_RESETLOGS_SCN
	SOURCE_RESETLOGS_TIME
	LOGMINER_ID
	NEGATIVE_RULE_SET_NAME
	NEGATIVE_RULE_SET_OWNER
	MAX_CHECKPOINT_SCN
	REQUIRED_CHECKPOINT_SCN
	LOGFILE_ASSIGNMENT
	STATUS_CHANGE_TIME
	ERROR_NUMBER
	ERROR_MESSAGE
	VERSION
	CAPTURE_TYPE
DBA_DIM_ATTRIBUTES	ATTRIBUTE_NAME
DBA_ERRORS	ATTRIBUTE
	MESSAGE_NUMBER
DBA_EXTERNAL_TABLES	PROPERTY
DBA_FGA_AUDIT_TRAIL	STATEMENT_TYPE
	EXTENDED_TIMESTAMP
	PROXY_SESSIONID
	GLOBAL_UID
	INSTANCE_NUMBER
	OS_PROCESS
	TRANSACTIONID
	STATEMENTID
	ENTRYID

Static Data Dictionary View	New Columns
DBA_INDEXES	IOT_REDUNDANT_PKEY_ELIM DROPPED
DBA_INDEXTYPES	ARRAY_DML
DBA_LOBS	TABLESPACE_NAME FORMAT PARTITIONED
DBA_LOG_GROUP_COLUMNS	LOGGING_PROPERTY
DBA_LOG_GROUPS	LOG_GROUP_TYPE GENERATED
DBA_LOGSTDBY_LOG	APPLIED
DBA_LOGSTDBY_PROGRESS	APPLIED_THREAD# APPLIED_SEQUENCE# READ_THREAD# READ_SEQUENCE# NEWEST_THREAD# NEWEST_SEQUENCE#
DBA_LOGSTDBY_SKIP	USE_LIKE ESC
DBA_LOGSTDBY_UNSUPPORTED	ATTRIBUTES TPROP SEGSPARE1 TFLAGS
DBA_MVIEWS	UNKNOWN_TRUSTED_FD STALE_SINCE
DBA_OBJ_AUDIT_OPTS	FBK
DBA_OBJECT_TABLES	DROPPED
DBA_OUTLINES	COMPATIBLE ENABLED FORMAT
DBA_PART_COL_STATISTICS	HISTOGRAM
DBA_POLICIES	IDX POLICY_TYPE LONG_PREDICATE
DBA_PROPAGATION	NEGATIVE_RULE_SET_OWNER NEGATIVE_RULE_SET_NAME
DBA_PROXIES	AUTHENTICATION
DBA_REGISTRY	NAMESPACE
DBA_REGISTRY_HIERARCHY	NAMESPACE

Static Data Dictionary View	New Columns
DBA_RSRC_PLAN_DIRECTIVES	MAX_IDLE_TIME MAX_IDLE_BLOCKER_TIME SWITCH_TIME_IN_CALL
DBA_SUBPART_COL_STATISTICS	HISTOGRAM
DBA_SUBSCRIBED_COLUMNS	SUBSCRIPTION_NAME
DBA_SUBSCRIBED_TABLES	SUBSCRIPTION_NAME
DBA_SUBSCRIPTIONS	SUBSCRIPTION_NAME
DBA_TAB_COL_STATISTICS	HISTOGRAM
DBA_TAB_COLS	HISTOGRAM QUALIFIED_COL_NAME
DBA_TAB_COLUMNS	HISTOGRAM
DBA_TAB_MODIFICATIONS	DROP_SEGMENTS
DBA_TABLES	DROPPED
DBA_TABLESPACES	RETENTION BIGFILE
USER_ALL_TABLES	DROPPED
USER_AUDIT_OBJECT	EXTENDED_TIMESTAMP PROXY_SESSIONID GLOBAL_UID INSTANCE_NUMBER OS_PROCESS TRANSACTIONID SCN SQL_BIND SQL_TEXT
USER_AUDIT_POLICIES	SEL INS UPD DEL AUDIT_TRAIL POLICY_COLUMN_OPTIONS
USER_AUDIT_SESSION	EXTENDED_TIMESTAMP PROXY_SESSIONID GLOBAL_UID INSTANCE_NUMBER OS_PROCESS

Static Data Dictionary View	New Columns
USER_AUDIT_STATEMENT	EXTENDED_TIMESTAMP PROXY_SESSIONID GLOBAL_UID INSTANCE_NUMBER OS_PROCESS TRANSACTIONID SCN SQL_BIND SQL_TEXT
USER_AUDIT_TRAIL	EXTENDED_TIMESTAMP PROXY_SESSIONID GLOBAL_UID INSTANCE_NUMBER OS_PROCESS TRANSACTIONID SCN SQL_BIND SQL_TEXT
USER_AW_PS	MAXPAGES
USER_DIM_ATTRIBUTES	ATTRIBUTE_NAME
USER_ERRORS	ATTRIBUTE MESSAGE_NUMBER
USER_EXTERNAL_TABLES	PROPERTY
USER_INDEXES	IOT_REDUNDANT_PKEY_ELIM DROPPED
USER_INDEXTYPES	ARRAY_DML
USER_LOBS	TABLESPACE_NAME FORMAT PARTITIONED
USER_LOG_GROUP_COLUMNS	LOGGING_PROPERTY
USER_LOG_GROUPS	LOG_GROUP_TYPE GENERATED
USER_MVIEWS	UNKNOWN_TRUSTED_FD STALE_SINCE
USER_OBJ_AUDIT_OPTS	FBK
USER_OBJECT_TABLES	DROPPED
USER_OUTLINES	COMPATIBLE ENABLED FORMAT
USER_PART_COL_STATISTICS	HISTOGRAM

Static Data Dictionary View	New Columns
USER_POLICIES	IDX POLICY_TYPE LONG_PREDICATE
USER_PROXIES	AUTHENTICATION
USER_PUBLISHED_COLUMNS	CHANGE_SET_NAME PUB_ID
USER_REGISTRY	NAMESPACE
USER_SUBPART_COL_STATISTICS	HISTOGRAM
USER_SUBSCRIBED_COLUMNS	SUBSCRIPTION_NAME
USER_SUBSCRIBED_TABLES	SUBSCRIPTION_NAME
USER_SUBSCRIPTIONS	SUBSCRIPTION_NAME
USER_TAB_COL_STATISTICS	HISTOGRAM
USER_TAB_COLS	HISTOGRAM QUALIFIED_COL_NAME
USER_TAB_COLUMNS	HISTOGRAM
USER_TAB_MODIFICATIONS	DROP_SEGMENTS
USER_TABLES	DROPPED
USER_TABLESPACES	RETENTION BIGFILE

See Also: [Chapter 2, "Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES"](#) and [Chapter 3, "Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS"](#) for more information about the new columns in these static data dictionary views

The following dynamic performance views have new columns in this release:

Dynamic Performance View	New Columns
GV\$ARCHIVE_DEST and V\$ARCHIVE_DEST	VALID_NOW VALID_TYPE VALID_ROLE DB_UNIQUE_NAME VERIFY EXPEDITE
GV\$ARCHIVE_DEST_STATUS and V\$ARCHIVE_DEST_STATUS	DB_UNIQUE_NAME
GV\$ARCHIVED_LOG and V\$ARCHIVED_LOG	RESETLOGS_ID IS_RECOVERY_DEST_FILE FAL

Dynamic Performance View	New Columns
GV\$AW_CALC and V\$AW_CALC	SESSION_ID CURR_DML_COMMAND PREV_DML_COMMAND AGGR_FUNC_LOGICAL_NA AGGR_FUNC_PRECOMPUTE AGGR_FUNC_CALCS
GV\$AW_SESSION_INFO and V\$AW_SESSION_INFO	SESSION_ID
GV\$BACKUP_ASYNC_IO and V\$BACKUP_ASYNC_IO	RMAN_STATUS_RECID RMAN_STATUS_STAMP
GV\$BACKUP_DATAFILE and V\$BACKUP_DATAFILE	USED_CHANGE_TRACKING BLOCKS_READ
GV\$BACKUP_PIECE and V\$BACKUP_PIECE	BYTES IS_RECOVERY_DEST_FILE RMAN_STATUS_RECID RMAN_STATUS_STAMP COMPRESSED
GV\$BACKUP_SYNC_IO and V\$BACKUP_SYNC_IO	RMAN_STATUS_RECID RMAN_STATUS_STAMP
GV\$BGPROCESS and V\$BGPROCESS	P SERIAL#
GV\$CLASS_CACHE_TRANSFER and V\$CLASS_CACHE_TRANSFER	CR_TRANSFER CURRENT_TRANSFER
GV\$CONTROLFILE and V\$CONTROLFILE	IS_RECOVERY_DEST_FILE
GV\$DATABASE and V\$DATABASE	SWITCHOVER# PLATFORM_ID PLATFORM_NAME RECOVERY_TARGET_INCARNATION# LAST_OPEN_INCARNATION# CURRENT_SCN FLASHBACK_ON SUPPLEMENTAL_LOG_DATA_FK SUPPLEMENTAL_LOG_DATA_ALL DB_UNIQUE_NAME STANDBY_BECAME_PRIMARY_SCN
GV\$DATABASE_INCARNATION and V\$DATABASE_INCARNATION	INCARNATION# STATUS RESETLOGS_ID PRIOR_INCARNATION#
GV\$DATAFILE_COPY and V\$DATAFILE_COPY	IS_RECOVERY_DEST_FILE RMAN_STATUS_RECID RMAN_STATUS_STAMP

Dynamic Performance View	New Columns
GV\$DB_CACHE_ADVICE and V\$DB_CACHE_ADVICE	ESTD_PHYSICAL_READ_TIME ESTD_PCT_OF_DB_TIME_FOR_READS
GV\$EVENT_NAME and V\$EVENT_NAME	EVENT_ID WAIT_CLASS_ID WAIT_CLASS# WAIT_CLASS
GV\$FAST_START_SERVERS and V\$FAST_START_SERVERS	XID
GV\$FAST_START_TRANSACTION and V\$FAST_START_TRANSACTIONS	XID PXID RCVSERVERS
GV\$INSTANCE_RECOVERY and V\$INSTANCE_RECOVERY	OPTIMAL_LOGFILE_SIZE ESTD_CLUSTER_AVAILABLE_TIME WRITES_MTTR WRITES_LOGFILE_SIZE WRITES_LOG_CHECKPOINT_SETTINGS WRITES_OTHER_SETTINGS WRITES_AUTOTUNE WRITES_FULL_THREAD_CKPT
GV\$LATCH and V\$LATCH	HASH
GV\$LATCH_CHILDREN and V\$LATCH_CHILDREN	HASH
GV\$LATCH_PARENT and V\$LATCH_PARENT	HASH
GV\$LATCHNAME and V\$LATCHNAME	HASH
GV\$LOADISTAT and V\$LOADISTAT	MESSAGE_NUM
GV\$LOG_HISTORY and V\$LOG_HISTORY	RESETLOGS_CHANGE# RESETLOGS_TIME
GV\$LOGFILE and V\$LOGFILE	IS_RECOVERY_DEST_FILE
GV\$LOGMNR_CONTENTS and V\$LOGMNR_CONTENTS	TABLE_NAME DATA_OBJV# SAFE_RESUME_SCN XID PXID AUDIT_SESSIONID
GV\$LOGMNR_DICTIONARY and V\$LOGMNR_DICTIONARY	DB_CREATED DB_VERSION_TIME DB_CHARACTER_SET DB_VERSION DB_STATUS DB_TXN_SCN

Dynamic Performance View	New Columns
GV\$LOGMNR_LOGS and V\$LOGMNR_LOGS	TYPE BLOCKSIZE FILESIZE
GV\$LOGMNR_PARAMETERS and V\$LOGMNR_PARAMETERS	REQUIRED_START_DATE REQUIRED_START_SCN OPTIONS
GV\$MANAGED_STANDBY and V\$MANAGED_STANDBY	RESETLOG_ID
GV\$OFFLINE_RANGE and V\$OFFLINE_RANGE	RESETLOGS_CHANGE# RESETLOGS_TIME
GV\$OPEN_CURSOR and V\$OPEN_CURSOR	SQL_ID
GV\$PARAMETER and V\$PARAMETER	DISPLAY_VALUE ISINSTANCE_MODIFIABLE ISDEPRECATED HASH
GV\$PARAMETER2 and V\$PARAMETER2	DISPLAY_VALUE ISINSTANCE_MODIFIABLE ISDEPRECATED
GV\$PROXY_ARCHIVEDLOG and V\$PROXY_ARCHIVEDLOG	TAG RMAN_STATUS_RECID RMAN_STATUS_STAMP
GV\$PROXY_DATAFILE and V\$PROXY_DATAFILE	RMAN_STATUS_RECID RMAN_STATUS_STAMP
GV\$RSRC_PLAN and V\$RSRC_PLAN	IS_TOP_PLAN

Dynamic Performance View	New Columns
GV\$SESSION and V\$SESSION	SQL_ID SQL_CHILD_NUMBER PREV_SQL_ID PREV_CHILD_NUMBER BLOCKING_SESSION_STATUS BLOCKING_SESSION SEQ# EVENT# EVENT P1TEXT P1 P1RAW P2TEXT P2 P2RAW P3TEXT P3 P3RAW WAIT_CLASS_ID WAIT_CLASS# WAIT_CLASS WAIT_TIME SECONDS_IN_WAIT STATE SERVICE_NAME
GV\$SESSION_EVENT and V\$SESSION_EVENT	EVENT_ID
GV\$SESSION_LONGOPS and V\$SESSION_LONGOPS	SQL_ID
GV\$SESSION_WAIT and V\$SESSION_WAIT	WAIT_CLASS_ID WAIT_CLASS# WAIT_CLASS
GV\$SGA_DYNAMIC_COMPONENTS and V\$SGA_DYNAMIC_COMPONENTS	USER_SPECIFIED_SIZE
GV\$SHARED_POOL_ADVICE and V\$SHARED_POOL_ADVICE	ESTD_LC_LOAD_TIME ESTD_LC_LOAD_TIME_FACTOR
GV\$SPPARAMETER and V\$SPPARAMETER	DISPLAY_VALUE

Dynamic Performance View	New Columns
GV\$SQL and V\$SQL	SQL_FULLTEXT SQL_ID END_OF_FETCH_COUNT DIRECT_WRITES APPLICATION_WAIT_TIME CONCURRENCY_WAIT_TIME CLUSTER_WAIT_TIME USER_IO_WAIT_TIME PLSQL_EXEC_TIME JAVA_EXEC_TIME OPTIMIZER_ENV OPTIMIZER_ENV_HASH_VALUE OLD_HASH_VALUE SQL_PROFILE PROGRAM_ID PROGRAM_LINE#
GV\$SQL_PLAN and V\$SQL_PLAN	SQL_ID PLAN_HASH_VALUE OBJECT_ALIAS OBJECT_TYPE PROJECTION TIME QBLOCK_NAME REMARKS
GV\$SQL_PLAN_STATISTICS and V\$SQL_PLAN_STATISTICS	SQL_ID PLAN_HASH_VALUE
GV\$SQL_PLAN_STATISTICS_ALL and V\$SQL_PLAN_STATISTICS_ALL	SQL_ID PLAN_HASH_VALUE OBJECT_ALIAS OBJECT_TYPE PROJECTION TIME QBLOCK_NAME REMARKS
GV\$SQL_REDIRECTION and V\$SQL_REDIRECTION	SQL_ID

Dynamic Performance View	New Columns
GV\$SQL_SHARED_CURSOR and V\$SQL_SHARED_CURSOR	SQL_ID CHILD_ADDRESS CHILD_NUMBER ANYDATA_TRANSFORMATION INCOMPLETE_CURSOR TOP_LEVEL_RPI_CURSOR DIFFERENT_LONG_LENGTH LOGICAL_STANDBY_APPLY DIFF_CALL_DURN BIND_UACS_DIFF PLSQL_CMP_SWITCHS_DIFF
GV\$SQL_SHARED_MEMORY and V\$SQL_SHARED_MEMORY	SQL_FULLTEXT SQL_ID
GV\$SQL_WORKAREA and V\$SQL_WORKAREA	SQL_ID
GV\$SQL_WORKAREA_ACTIVE and V\$SQL_WORKAREA_ACTIVE	SQL_HASH_VALUE SQL_ID
GV\$SQLAREA and V\$SQLAREA	SQL_ID END_OF_FETCH_COUNT DIRECT_WRITES APPLICATION_WAIT_TIME CONCURRENCY_WAIT_TIME CLUSTER_WAIT_TIME USER_IO_WAIT_TIME PLSQL_EXEC_TIME JAVA_EXEC_TIME OLD_HASH_VALUE PROGRAM_ID
GV\$SQLTEXT and V\$SQLTEXT	SQL_ID
GV\$SQLTEXT_WITH_NEWLINES and V\$SQLTEXT_WITH_NEWLINES	SQL_ID
GV\$STANDBY_LOG and V\$STANDBY_LOG	DBID
GV\$STATNAME and V\$STATNAME	STAT_ID
GV\$STREAMS_APPLY_COORDINATOR and V\$STREAMS_APPLY_COORDINATOR	TOTAL_IGNORED TOTAL_ROLLBACKS
GV\$STREAMS_APPLY_READER and V\$STREAMS_APPLY_READER	LAST_BROWSE_NUM OLDEST_SCN_NUM LAST_BROWSE_SEQ LAST_DEQ_SEQ
GV\$STREAMS_APPLY_SERVER and V\$STREAMS_APPLY_SERVER	TOTAL_ROLLBACKS

Dynamic Performance View	New Columns
GV\$STREAMS_CAPTURE and V\$STREAMS_CAPTURE	LOGMINER_ID TOTAL_PREFILTER_DISCARDED TOTAL_PREFILTER_KEPT TOTAL_PREFILTER_EVALUATIONS TOTAL_MESSAGES_CREATED TOTAL_FULL_EVALUATIONS AVAILABLE_MESSAGE_NUMBER AVAILABLE_MESSAGE_CREATE_TIME ELAPSED_REDO_WAIT_TIME ELAPSED_PAUSE_TIME
GV\$SYSSTAT and V\$SYSSTAT	STAT_ID
GV\$SYSTEM_EVENT and V\$SYSTEM_EVENT	EVENT_ID
GV\$SYSTEM_PARAMETER and V\$SYSTEM_PARAMETER	DISPLAY_VALUE ISINSTANCE_MODIFIABLE ISDEPRECATED HASH
GV\$SYSTEM_PARAMETER2 and V\$SYSTEM_PARAMETER2	DISPLAY_VALUE ISINSTANCE_MODIFIABLE ISDEPRECATED
GV\$TABLESPACE and V\$TABLESPACE	BIGFILE FLASHBACK_ON
GV\$TEMPORARY_LOBS and V\$TEMPORARY_LOBS	ABSTRACT_LOBS
GV\$TEMPSEG_USAGE and V\$TEMPSEG_USAGE	SQL_ID
GV\$TRANSACTION and V\$TRANSACTION	START_DATE DSCN_BASE DSCN_WRAP START_SCN DEPENDENT_SCN XID PRV_XID PTX_XID
GV\$UNDOSTAT and V\$UNDOSTAT	MAXQUERYID ACTIVEBLKS UNEXPIREDBLKS EXPIREDBLKS TUNED_UNDORETENTION
GV\$VPD_POLICY and V\$VPD_POLICY	SQL_ID

See Also: [Chapter 4, "Dynamic Performance \(V\\$\) Views"](#) for more information about the new columns in these dynamic performance views

Part I

Initialization Parameters

This part describes the database initialization parameters that can be specified in a parameter file to start or configure an instance. This part contains the following chapter:

- [Chapter 1, "Initialization Parameters"](#)

Initialization Parameters

This chapter contains detailed descriptions (in alphabetical order) of the database initialization parameters.

This chapter contains the following topics:

- [Uses of Initialization Parameters](#)
- [Basic Initialization Parameters](#)
- [Parameter Files](#)
- [Changing Parameter Values in a Parameter File](#)
- [Reading the Parameter Descriptions](#)
- [Initialization Parameter Descriptions](#)

Uses of Initialization Parameters

Initialization parameters fall into various functional groups. For example, parameters perform the following functions:

- Set limits for the entire database
- Set user or process limits
- Set limits on database resources
- Affect performance (these are called **variable parameters**)

Variable parameters are of particular interest to database administrators, because these parameters are used primarily to improve database performance.

Database administrators can use initialization parameters to:

- Optimize performance by adjusting memory structures, such as the number of database buffers in memory
- Set database-wide defaults, such as the amount of space initially allocated for a context area when it is created
- Set database limits, such as the maximum number of database users
- Specify names of files or directories required by the database

Many initialization parameters can be fine-tuned to improve database performance. Other parameters should never be altered or should be altered only under the supervision of Oracle Support Services.

All initialization parameters are optional. Oracle has a default value for each parameter. This value may be operating system-dependent, depending on the parameter.

Types of Initialization Parameters

The Oracle database server has the following types of initialization parameters:

- [Derived Parameters](#)
- [Operating System-Dependent Parameters](#)
- [Variable Parameters](#) (these can be dynamic parameters or any of the preceding ones)

Derived Parameters

Some initialization parameters are **derived**, meaning that their values are calculated from the values of other parameters. Normally, you should not alter values for derived parameters, but if you do, then the value you specify will override the calculated value.

For example, the default value of the `SESSIONS` parameter is derived from the value of the `PROCESSES` parameter. If the value of `PROCESSES` changes, then the default value of `SESSIONS` changes as well, unless you override it with a specified value.

Operating System-Dependent Parameters

The valid values or value ranges of some initialization parameters depend upon the host operating system. For example, the parameter `DB_BLOCK_BUFFERS` indicates the number of data buffers in main memory, and its maximum value depends on the operating system. The size of those buffers, set by `DB_BLOCK_SIZE`, has an operating system-dependent default value.

See Also: Your operating system-specific Oracle documentation for more information on operating system-dependent Oracle parameters and operating system parameters

Variable Parameters

The variable initialization parameters offer the most potential for improving system performance. Some variable parameters set capacity limits but do not affect performance. For example, when the value of `OPEN_CURSORS` is 10, a user process attempting to open its eleventh cursor receives an error. Other variable parameters affect performance but do not impose absolute limits. For example, reducing the value of `DB_BLOCK_BUFFERS` does not prevent work even though it may slow down performance.

Increasing the values of variable parameters may improve your system's performance, but increasing most parameters also increases the system global area (SGA) size. A larger SGA can improve database performance up to a point. In virtual memory operating systems, an SGA that is too large can degrade performance if it is swapped in and out of memory. Operating system parameters that control virtual memory working areas should be set with the SGA size in mind. The operating system configuration can also limit the maximum size of the SGA.

Basic Initialization Parameters

The following is a list of the database basic initialization parameters. Most databases should only need to have basic parameters set to run properly and efficiently. Oracle advises you to become familiar with the basic parameters and only use other parameters when directed to by feature documentation or in special circumstances:

```

CLUSTER_DATABASE
COMPATIBLE
CONTROL_FILES
DB_BLOCK_SIZE
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_DOMAIN
DB_NAME
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_UNIQUE_NAME
INSTANCE_NUMBER
JOB_QUEUE_PROCESSES
LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n
NLS_LANGUAGE
NLS_TERRITORY
OPEN_CURSORS
PGA_AGGREGATE_TARGET
PROCESSES
REMOTE_LISTENER
REMOTE_LOGIN_PASSWORDFILE
ROLLBACK_SEGMENTS
SESSIONS
SGA_TARGET
SHARED_SERVERS
STAR_TRANSFORMATION_ENABLED
UNDO_MANAGEMENT
UNDO_TABLESPACE

```

Parameter Files

A **parameter file** is a file that contains a list of initialization parameters and a value for each parameter. You specify initialization parameters in a parameter file that reflect your particular installation. Oracle supports the following two types of parameter files:

- [Server Parameter Files](#)
- [Initialization Parameter Files](#)

Server Parameter Files

A **server parameter file** is a binary file that acts as a repository for initialization parameters. The server parameter file can reside on the machine where the Oracle database server executes. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.

Initialization Parameter Files

An **initialization parameter file** is a text file that contains a list of initialization parameters. The file should be written in the client's default character set.

The following are sample entries in an initialization parameter file:

```
PROCESSES = 100
OPEN_LINKS = 12
GLOBAL_NAMES = true
```

The name of the initialization parameter file varies depending on the operating system. For example, it can be in mixed case or lowercase, or it can have a logical name or a variation of the name `init.ora`. Also supplied is an `initdw.ora` file, which contains suggested parameter settings for data warehouses and data marts. The database administrator can choose a different filename for the initialization parameter file.

Refer to your operating system-specific Oracle documentation for the default locations and filenames for initialization parameter files on your operating system. The initialization parameter file is read by the client-side tool used to start the server (such as SQL*Plus).

Sample initialization parameter files are provided on the Oracle distribution medium for each operating system. A sample file is sufficient for initial use, but you will probably want to modify the file to tune the database for best performance. Any changes will take effect after you completely shut down and restart the instance.

Specifying Values in an Initialization Parameter File

This section describes several aspects of setting parameter values in an initialization parameter file.

Rules Governing Initialization Parameter Files The following rules govern the specification of parameters in initialization parameter files:

- An initialization parameter file should contain only parameters and comments. A pound sign (#) starts a comment line. The rest of the line is ignored.
- You can specify parameters in any order.
- Case (upper or lower) in filenames is significant only if case is significant on the host operating system.
- To enter several parameters on one line, use spaces between parameter names and values, as in the following example:

```
PROCESSES = 100 CPU_COUNT = 1 OPEN_CURSORS = 10
```

- Some parameters, such as `ROLLBACK_SEGMENTS`, accept multiple value entries. Enter multiple values enclosed in parentheses and separated by commas. For example:

```
ROLLBACK_SEGMENTS = (SEG1, SEG2, SEG3, SEG4, SEG5)
```

Alternatively, you can enter multiple values without parentheses and commas. For example:

```
ROLLBACK_SEGMENTS = SEG1 SEG2 SEG3 SEG4 SEG5
```

Either syntax is valid.

If you enter values for one parameter in multiple entries, then the entries must be on consecutive lines. If they are not, then the first entry will not be processed properly. For example, in the following entry the setting for `SEG3` and `SEG4` will override the setting for `SEG1` and `SEG2`:

```
ROLLBACK_SEGMENTS = SEG1 SEG2
OPEN_CURSORS = 10
ROLLBACK_SEGMENTS = SEG3 SEG4
```

- A backslash (`\`), also known as an escape character, indicates continuation of the parameter specification. If a backslash continues a line, then the continued line must have no leading spaces. For example:

```
ROLLBACK_SEGMENTS = (SEG1, SEG2, \
SEG3, SEG4, SEG5)
```

- You can use the `IFILE` initialization parameter to embed the contents of another initialization parameter file into the current initialization parameter file.
- Enclose in quotation marks any parameter values that contain spaces or tabs. You can use either single or double quotation marks unless otherwise indicated. For example:

```
NLS_TERRITORY = 'CZECH REPUBLIC'
```

Note: Listing parameters in alphabetical order in the initialization parameter file can help you to find them and can help ensure that each parameter is specified only once.

- Enclose in quotation marks any parameter value that contains a special character.

See Also:

- Your operating system-specific Oracle documentation for more information on initialization parameter files
- ["IFILE"](#) on page 1-60

Using Special Characters in Parameter Values If a parameter value contains a special character, then the special character must be preceded by a backslash or the entire parameter value must be enclosed in quotation marks. For example, you can specify special characters using either of the following:

```
DB_DOMAIN = 'JAPAN.ACME#.COM'
```

```
DB_DOMAIN = JAPAN.ACME\#.COM
```

[Table 1–1](#) lists the special characters you can use in initialization parameter files.

Table 1–1 Special Characters in Initialization Parameter Files

Character	Name	Description
#	Number sign	Comment
(Left parenthesis	Start list of values
)	Right parenthesis	End list of values

Table 1–1 (Cont.) Special Characters in Initialization Parameter Files

Character	Name	Description
"	Double quotation mark	Start or end of quoted string
'	Single quotation mark	Start or end of quoted string
=	Equal sign	Separator of keyword and values
,	Comma	Separator of elements
-	Minus sign	Precedes UNIX-style keywords
\	Backslash	Escape character

If a special character must be treated literally in an initialization parameter file, then it must either be preceded by the backslash character, or the entire string containing the special character must be enclosed in quotation marks.

Using the Escape Character As described in "[Rules Governing Initialization Parameter Files](#)" on page 1-4, the backslash (\) can also signify a line continuation. If the backslash is followed by an alphanumeric character, then the backslash is treated as a normal character in the input. If it is not followed by an alphanumeric character, then the backslash is treated either as a backslash or as a continuation character.

Using Quotation Marks Quotation marks can be nested in one of two ways. The first method is to double the quotation marks in the nested string. For example:

```
NLS_DATE_FORMAT = '''Today is'' MM/DD/YYYY'
```

The second method is to alternate single and double quotation marks. For example:

```
NLS_DATE_FORMAT = '"Today is" MM/DD/YYYY'
```

Changing Parameter Values in a Parameter File

You change the value of a parameter in a parameter file in one of the following ways:

- By editing an initialization parameter file
 - In most cases, the new value takes effect the next time you start an instance of the database.
- By issuing an `ALTER SYSTEM SET ... SCOPE=SPFILE` statement to update a server parameter file

Parameters by Functional Category

The following list shows the initialization parameters by their functional category:

- **ANSI Compliance**
 - BLANK_TRIMMING
- **Backup and Restore**
 - BACKUP_TAPE_IO_SLAVES
 - TAPE_ASYNC_IO
- **BFILEs**

SESSION_MAX_OPEN_FILES

- **Buffer Cache and I/O**

DB_nK_CACHE_SIZE
DB_BLOCK_SIZE
DB_CACHE_ADVICE
DB_CACHE_SIZE
DB_FILE_MULTIBLOCK_READ_COUNT
DB_KEEP_CACHE_SIZE
DB_RECYCLE_CACHE_SIZE
DB_WRITER_PROCESSES
DBWR_IO_SLAVES
DISK_ASYNC_IO
FILESYSTEMIO_OPTIONS
READ_ONLY_OPEN_DELAYED
USE_INDIRECT_DATA_BUFFERS

- **Cursors and Library Cache**

CURSOR_SHARING
CURSOR_SPACE_FOR_TIME
OPEN_CURSORS
SERIAL_REUSE
SESSION_CACHED_CURSORS

- **Database/Instance Identification**

DB_DOMAIN
DB_NAME
INSTANCE_NAME

- **Diagnostics and Statistics**

BACKGROUND_CORE_DUMP
BACKGROUND_DUMP_DEST
CORE_DUMP_DEST
DB_BLOCK_CHECKING
DB_BLOCK_CHECKSUM
EVENT
MAX_DUMP_FILE_SIZE
SHADOW_CORE_DUMP
SQL_TRACE
STATISTICS_LEVEL
TIMED_OS_STATISTICS
TIMED_STATISTICS
TRACE_ENABLED
TRACEFILE_IDENTIFIER
USER_DUMP_DEST

- **Distributed, Replication**

COMMIT_POINT_STRENGTH
DISTRIBUTED_LOCK_TIMEOUT
GLOBAL_NAMES
HS_AUTOREGISTER
OPEN_LINKS
OPEN_LINKS_PER_INSTANCE

REPLICATION_DEPENDENCY_TRACKING

- **File Locations, Names, and Sizes**

AUDIT_FILE_DEST
BACKGROUND_CORE_DUMP
BACKGROUND_DUMP_DEST
CONTROL_FILES
CORE_DUMP_DEST
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_FILES
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
FILE_MAPPING
IFILE
LOG_ARCHIVE_DEST_n
SPFILE

- **Globalization**

NLS_CALENDAR
NLS_COMP
NLS_CURRENCY
NLS_DATE_FORMAT
NLS_DATE_LANGUAGE
NLS_DUAL_CURRENCY
NLS_ISO_CURRENCY
NLS_LANGUAGE
NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP
NLS_NUMERIC_CHARACTERS
NLS_SORT
NLS_TERRITORY
NLS_TIMESTAMP_FORMAT
NLS_TIMESTAMP_TZ_FORMAT

- **Java**

JAVA_MAX_SESSIONSPACE_SIZE
JAVA_POOL_SIZE
JAVA_SOFT_SESSIONSPACE_LIMIT

- **Job Queues**

JOB_QUEUE_PROCESSES

- **License Limits**

LICENSE_MAX_SESSIONS
LICENSE_MAX_USERS
LICENSE_SESSIONS_WARNING

- **Miscellaneous**

AQ_TM_PROCESSES
COMPATIBLE
FIXED_DATE

LOGMNR_MAX_PERSISTENT_SESSIONS

- **Networking**

LOCAL_LISTENER
REMOTE_LISTENER
SERVICE_NAMES

- **Objects and LOBs**

OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE

- **OLAP**

OLAP_PAGE_POOL_SIZE

- **Optimizer**

OPTIMIZER_DYNAMIC_SAMPLING
OPTIMIZER_FEATURES_ENABLE
OPTIMIZER_INDEX_CACHING
OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_MODE
QUERY_REWRITE_ENABLED
QUERY_REWRITE_INTEGRITY
STAR_TRANSFORMATION_ENABLED

- **Parallel Execution**

PARALLEL_ADAPTIVE_MULTI_USER
PARALLEL_AUTOMATIC_TUNING
PARALLEL_EXECUTION_MESSAGE_SIZE
PARALLEL_MAX_SERVERS
PARALLEL_MIN_PERCENT
PARALLEL_MIN_SERVERS
PARALLEL_THREADS_PER_CPU

- **PL/SQL**

PLSQL_CODE_TYPE
PLSQL_DEBUG
PLSQL_NATIVE_LIBRARY_DIR
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
PLSQL_OPTIMIZE_LEVEL
PLSQL_V2_COMPATIBILITY
PLSQL_WARNINGS
REMOTE_DEPENDENCIES_MODE
UTL_FILE_DIR

- **Memory**

LARGE_POOL_SIZE
SHARED_POOL_RESERVED_SIZE
SHARED_POOL_SIZE

- **SGA Memory**

DB_nK_CACHE_SIZE

DB_CACHE_ADVICE
DB_CACHE_SIZE
HI_SHARED_MEMORY_ADDRESS
JAVA_POOL_SIZE
LOCK_SGA
OLAP_PAGE_POOL_SIZE
PGA_AGGREGATE_TARGET
PRE_PAGE_SGA
SGA_MAX_SIZE
SHARED_MEMORY_ADDRESS

■ **Real Application Clusters**

ACTIVE_INSTANCE_COUNT
CLUSTER_DATABASE
CLUSTER_DATABASE_INSTANCES
CLUSTER_INTERCONNECTS
GC_FILES_TO_LOCKS
INSTANCE_GROUPS
INSTANCE_NUMBER
LOCK_NAME_SPACE
MAX_COMMIT_PROPAGATION_DELAY
PARALLEL_INSTANCE_GROUP
THREAD

■ **Redo Logs, Archiving, and Recovery**

CONTROL_FILE_RECORD_KEEP_TIME
DB_CREATE_ONLINE_LOG_DEST_n
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
FAST_START_IO_TARGET
FAST_START_MTTR_TARGET
LOG_BUFFER
LOG_CHECKPOINT_INTERVAL
LOG_CHECKPOINT_TIMEOUT
LOG_CHECKPOINTS_TO_ALERT
LOG_ARCHIVE_CONFIG
LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n
LOG_ARCHIVE_DUPLEX_DEST
LOG_ARCHIVE_FORMAT
LOG_ARCHIVE_MAX_PROCESSES
LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_ARCHIVE_TRACE
RECOVERY_PARALLELISM

■ **Resource Manager**

RESOURCE_LIMIT
RESOURCE_MANAGER_PLAN

■ **Security and Auditing**

AUDIT_FILE_DEST
AUDIT_SYS_OPERATIONS
AUDIT_TRAIL

O7_DICTIONARY_ACCESSIBILITY
OS_AUTHENT_PREFIX
OS_ROLES
RDBMS_SERVER_DN
REMOTE_LOGIN_PASSWORDFILE
REMOTE_OS_AUTHENT
REMOTE_OS_ROLES
SQL92_SECURITY

- **Sessions and Processes**

CPU_COUNT
PROCESSES
SESSIONS

- **Shared Server Architecture**

CIRCUITS
DISPATCHERS
MAX_DISPATCHERS
MAX_SHARED_SERVERS
SHARED_SERVER_SESSIONS
SHARED_SERVERS

- **Standby Database**

ARCHIVE_LAG_TARGET
DB_FILE_NAME_CONVERT
DB_UNIQUE_NAME
DG_BROKER_CONFIG_FILEn
DG_BROKER_START
FAL_CLIENT
FAL_SERVER
LOG_FILE_NAME_CONVERT
REMOTE_ARCHIVE_ENABLE
STANDBY_ARCHIVE_DEST
STANDBY_FILE_MANAGEMENT

- **Undo Management**

RESUMABLE_TIMEOUT
ROLLBACK_SEGMENTS
TRANSACTIONS_PER_ROLLBACK_SEGMENT
UNDO_MANAGEMENT
UNDO_RETENTION
UNDO_TABLESPACE

- **Temporary Sort Space**

BITMAP_MERGE_AREA_SIZE
CREATE_BITMAP_AREA_SIZE
HASH_AREA_SIZE
PGA_AGGREGATE_TARGET
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
WORKAREA_SIZE_POLICY

- **Transactions**

DML_LOCKS
 ENQUEUE_RESOURCES
 FAST_START_PARALLEL_ROLLBACK
 TRANSACTIONS

Modifiable Parameters

Some initialization parameters can be modified using the ALTER SESSION or ALTER SYSTEM statements while an instance is running. Use the following syntax to modify initialization parameters:

```
ALTER SESSION SET parameter_name = value
ALTER SYSTEM SET parameter_name = value [DEFERRED]
```

Whenever a parameter is modified using the ALTER SYSTEM statement, the Oracle Database records the statement that modifies the parameter in the alert log.

The ALTER SESSION statement changes the value of the specified parameter for the duration of the session that invokes the statement. The value of the parameter does not change for other sessions in the instance. The value of the following initialization parameters can be changed with ALTER SESSION:

ASM_POWER_LIMIT
 CREATE_STORED_OUTLINES
 CURSOR_SHARING
 DB_BLOCK_CHECKING
 DB_CREATE_FILE_DEST
 DB_CREATE_ONLINE_LOG_DEST_1
 DB_FILE_MULTIBLOCK_READ_COUNT
 DB_FILE_NAME_CONVERT
 DDL_WAIT_FOR_LOCKS
 FILESYSTEMIO_OPTIONS
 GLOBAL_NAMES
 HASH_AREA_SIZE
 LOG_ARCHIVE_DEST_1
 LOG_ARCHIVE_DEST_STATE_1
 LOG_ARCHIVE_MIN_SUCCEED_DEST
 MAX_DUMP_FILE_SIZE
 NLS_CALENDAR
 NLS_COMP
 NLS_CURRENCY
 NLS_DATE_FORMAT
 NLS_DATE_LANGUAGE
 NLS_DUAL_CURRENCY
 NLS_ISO_CURRENCY
 NLS_LANGUAGE
 NLS_LENGTH_SEMANTICS
 NLS_NCHAR_CONV_EXCP
 NLS_NUMERIC_CHARACTERS
 NLS_SORT
 NLS_TERRITORY
 NLS_TIMESTAMP_FORMAT
 NLS_TIMESTAMP_TZ_FORMAT
 OBJECT_CACHE_MAX_SIZE_PERCENT
 OBJECT_CACHE_OPTIMAL_SIZE
 OLAP_PAGE_POOL_SIZE

OPTIMIZER_DYNAMIC_SAMPLING
 OPTIMIZER_FEATURES_ENABLE
 OPTIMIZER_INDEX_CACHING
 OPTIMIZER_INDEX_COST_ADJ
 OPTIMIZER_MODE
 PARALLEL_INSTANCE_GROUP
 PARALLEL_MIN_PERCENT
 PLSQL_CODE_TYPE
 PLSQL_DEBUG
 PLSQL_OPTIMIZE_LEVEL
 PLSQL_V2_COMPATIBILITY
 PLSQL_WARNINGS
 QUERY_REWRITE_ENABLED
 QUERY_REWRITE_INTEGRITY
 REMOTE_DEPENDENCIES_MODE
 RESUMABLE_TIMEOUT
 SESSION_CACHED_CURSORS
 SKIP_UNUSABLE_INDEXES
 SORT_AREA_RETAINED_SIZE
 SORT_AREA_SIZE
 SQL_TRACE
 SQTUNE_CATEGORY
 STAR_TRANSFORMATION_ENABLED
 STATISTICS_LEVEL
 TIMED_OS_STATISTICS
 TIMED_STATISTICS
 TRACEFILE_IDENTIFIER
 WORKAREA_SIZE_POLICY

The ALTER SYSTEM statement without the DEFERRED keyword modifies the global value of the parameter for all sessions in the instance, for the duration of the instance (until the database is shut down). The value of the following initialization parameters can be changed with ALTER SYSTEM:

AQ_TM_PROCESSES
 ARCHIVE_LAG_TARGET
 ASM_DISKGROUPS
 ASM_DISKSTRING
 ASM_POWER_LIMIT
 BACKGROUND_DUMP_DEST
 CIRCUITS
 CONTROL_FILE_RECORD_KEEP_TIME
 CORE_DUMP_DEST
 CPU_COUNT
 CREATE_STORED_OUTLINES
 CURSOR_SHARING
 DB_nK_CACHE_SIZE
 DB_BLOCK_CHECKING
 DB_BLOCK_CHECKSUM
 DB_CACHE_ADVICE
 DB_CACHE_SIZE
 DB_CREATE_FILE_DEST
 DB_CREATE_ONLINE_LOG_DEST_n
 DB_FILE_MULTIBLOCK_READ_COUNT
 DB_FLASHBACK_RETENTION_TARGET
 DB_KEEP_CACHE_SIZE

DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_RECYCLE_CACHE_SIZE
DDL_WAIT_FOR_LOCKS
DG_BROKER_CONFIG_FILE*n*
DG_BROKER_START
DISPATCHERS
FAL_CLIENT
FAL_SERVER
FAST_START_IO_TARGET
FAST_START_MTTR_TARGET
FAST_START_PARALLEL_ROLLBACK
FILE_MAPPING
FILESYSTEMIO_OPTIONS
FIXED_DATE
GLOBAL_NAMES
HS_AUTOREGISTER
JAVA_POOL_SIZE
JOB_QUEUE_PROCESSES
LARGE_POOL_SIZE
LDAP_DIRECTORY_ACCESS
LICENSE_MAX_SESSIONS
LICENSE_MAX_USERS
LICENSE_SESSIONS_WARNING
LOCAL_LISTENER
LOG_ARCHIVE_CONFIG
LOG_ARCHIVE_DEST
LOG_ARCHIVE_DEST_*n*
LOG_ARCHIVE_DEST_STATE_*n*
LOG_ARCHIVE_DUPLEX_DEST
LOG_ARCHIVE_LOCAL_FIRST
LOG_ARCHIVE_MAX_PROCESSES
LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_ARCHIVE_TRACE
LOG_CHECKPOINT_INTERVAL
LOG_CHECKPOINT_TIMEOUT
LOG_CHECKPOINTS_TO_ALERT
MAX_DISPATCHERS
MAX_DUMP_FILE_SIZE
MAX_SHARED_SERVERS
NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP
OPEN_CURSORS
OPTIMIZER_DYNAMIC_SAMPLING
OPTIMIZER_FEATURES_ENABLE
OPTIMIZER_INDEX_CACHING
OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_MODE
PARALLEL_ADAPTIVE_MULTI_USER
PARALLEL_INSTANCE_GROUP
PARALLEL_MAX_SERVERS
PARALLEL_MIN_SERVERS
PARALLEL_THREADS_PER_CPU
PGA_AGGREGATE_TARGET
PLSQL_CODE_TYPE

```
PLSQL_DEBUG
PLSQL_NATIVE_LIBRARY_DIR
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
PLSQL_OPTIMIZE_LEVEL
PLSQL_V2_COMPATIBILITY
PLSQL_WARNINGS
QUERY_REWRITE_ENABLED
QUERY_REWRITE_INTEGRITY
REMOTE_DEPENDENCIES_MODE
REMOTE_LISTENER
RESOURCE_LIMIT
RESOURCE_MANAGER_PLAN
RESUMABLE_TIMEOUT
SERVICE_NAMES
SGA_TARGET
SHARED_POOL_SIZE
SHARED_SERVER_SESSIONS
SHARED_SERVERS
SKIP_UNUSABLE_INDEXES
SQL_TRACE
SQLTUNE_CATEGORY
STANDBY_ARCHIVE_DEST
STANDBY_FILE_MANAGEMENT
STAR_TRANSFORMATION_ENABLED
STATISTICS_LEVEL
STREAMS_POOL_SIZE
TIMED_OS_STATISTICS
TIMED_STATISTICS
TRACE_ENABLED
UNDO_RETENTION
UNDO_TABLESPACE
USER_DUMP_DEST
WORKAREA_SIZE_POLICY
```

The `ALTER SYSTEM . . . DEFERRED` statement does not modify the global value of the parameter for existing sessions, but the value will be modified for future sessions that connect to the database. The value of the following initialization parameters can be changed with `ALTER SYSTEM . . . DEFERRED`:

```
AUDIT_FILE_DEST
BACKUP_TAPE_IO_SLAVES
OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE
OLAP_PAGE_POOL_SIZE
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
```

Displaying Current Parameter Values

To see the current settings for initialization parameters, use the following SQL*Plus command:

```
SQL> SHOW PARAMETERS
```

This command displays all parameters in alphabetical order, along with their current values.

Enter the following text string to display all parameters having BLOCK in their names:

```
SQL> SHOW PARAMETERS BLOCK
```

You can use the SPOOL command to write the output to a file.

Parameters You Should Not Specify in the Parameter File

You should not specify the following two types of parameters in your parameter files:

- Parameters that you never alter except when instructed to do so by Oracle to resolve a problem
- Derived parameters, which normally do not need altering because their values are calculated automatically by the Oracle database server

When Parameters Are Set Incorrectly

Some parameters have a minimum setting below which an Oracle instance will not start. For other parameters, setting the value too low or too high may cause Oracle to perform badly, but it will still run. Also, Oracle may convert some values outside the acceptable range to usable levels.

If a parameter value is too low or too high, or you have reached the maximum for some resource, then Oracle returns an error. Frequently, you can wait a short while and retry the operation when the system is not as busy. If a message occurs repeatedly, then you should shut down the instance, adjust the relevant parameter, and restart the instance.

Reading the Parameter Descriptions

The parameter descriptions in this chapter adhere to the following format.

PARAMETER_NAME

Property	Description
Parameter type	<p>Specifies the type of the parameter:</p> <ul style="list-style-type: none"> ■ A Boolean parameter accepts either <code>true</code> or <code>false</code> as its value. ■ A string parameter accepts any sequence of characters as its value, subject to the syntax for the parameter. ■ An integer parameter accepts a 4-byte value that can range from 0 to $2^{32} - 1$. ■ A parameter file parameter accepts an initialization parameter file specification as its value. ■ A big integer parameter accepts an 8-byte value that can range from 0 to $2^{64} - 1$. You specify a value for a big integer as an integer together with an optional modifier such as K, M, or G, which respectively denotes kilobytes, megabytes, or gigabytes. For example, 1000, 100 KB, 50 MB and 2 GB are valid specifications for big integers.
Syntax	For string and big integer parameters, specifies the valid syntax for specifying the parameter.

Property	Description
Default value	Specifies the value this parameter assumes if not explicitly specified.
Modifiable	Specifies whether the parameter can be changed for the current session (by an <code>ALTER SESSION</code> statement) or for all sessions in the current instance (by an <code>ALTER SYSTEM</code> statement): <ul style="list-style-type: none"> ▪ <code>ALTER SESSION</code> overrides the instance-wide setting of the parameter for the current session only. You can restore the instance-wide setting for that session only by issuing another <code>ALTER SESSION</code> statement. ▪ <code>ALTER SYSTEM</code> can be used to change the value in the server parameter file (SPFILE) of any initialization parameter. Such a change takes effect only in subsequent instances. The parameter descriptions indicate only those parameters that can be modified for the current instance.
Range of values	Specifies the valid range of values that this parameter can assume, shown as a minimum and maximum value. Not applicable to all parameters.
Real Application Clusters	Specifies how the values for this parameter must be specified for multiple instances in a cluster database environment. Not applicable to all parameters.

For each parameter, paragraphs following these details further describe the parameter and the effects of different settings.

Initialization Parameter Descriptions

The remainder of this chapter describes the initialization parameters in alphabetical order. Initialization parameter values apply to the entire database, not to an individual user, unless otherwise specified.

Note: Parameters that have become obsolete are not documented.

See Also:

- *Oracle Database Upgrade Guide* for information about obsolete parameters
- Your system release bulletins or other operating system-specific Oracle documentation

ACTIVE_INSTANCE_COUNT

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	No
Range of values	1 or \geq the number of instances in the cluster. (Values other than 1 have no effect on the active or standby status of any instances.)

Property	Description
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

Note: This parameter functions only in a cluster with only two instances.

ACTIVE_INSTANCE_COUNT enables you to designate one instance in a two-instance cluster as the primary instance and the other instance as the secondary instance. This parameter has no functionality in a cluster with more than two instances.

When you set this parameter to 1, the first instance you start up becomes the primary instance and accepts client connections. The second instance starts up as a secondary instance and can accept client connections only if the first instance fails. In such an event, the secondary instance becomes the primary instance.

When the failed instance can once again be started up, it starts up as the secondary instance, and will not accept client connections unless the current primary instance fails.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide* for more information on setting this parameter

AQ_TM_PROCESSES

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to 10
Basic	No

AQ_TM_PROCESSES enables time monitoring of queue messages. The times can be used in messages that specify delay and expiration properties. Values from 1 to 10 specify the number of queue monitor processes created to monitor the messages. If AQ_TM_PROCESSES is not specified or is set to 0, then the queue monitor is not created.

See Also: *Oracle Streams Advanced Queuing User's Guide and Reference* for more information about this parameter and Advanced Queuing

ARCHIVE_LAG_TARGET

Property	Description
Parameter type	Integer
Default value	0 (disabled)

Property	Description
Modifiable	ALTER SYSTEM
Range of values	0 or any integer in [60, 7200]
Basic	No
Real Application Clusters	Multiple instances should use the same value.

ARCHIVE_LAG_TARGET limits the amount of data that can be lost and effectively increases the availability of the standby database by forcing a log switch after the specified amount of time elapses.

A 0 value disables the time-based thread advance feature; otherwise, the value represents the number of seconds. Values larger than 7200 seconds are not of much use in maintaining a reasonable lag in the standby database. The typical, or recommended value is 1800 (30 minutes). Extremely low values can result in frequent log switches, which could degrade performance; such values can also make the archiver process too busy to archive the continuously generated logs.

See Also: *Oracle Data Guard Concepts and Administration* for more information about criteria for setting this parameter

ASM_DISKGROUPS

Property	Description
Parameter type	String
Syntax	ASM_DISKGROUPS = <i>diskgroup</i> [, <i>diskgroup</i>] ...
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	Comma-separated list of strings, of up to 30 characters
Real Application Clusters	Multiple instances can have different values.

Note: This parameter may only be specified in an Automatic Storage Management instance.

ASM_DISKGROUPS specifies a list of names of disk groups to be mounted by an Automatic Storage Management instance at instance startup or when an ALTER DISKGROUP ALL MOUNT statement is issued.

Automatic Storage Management automatically adds a disk group to this parameter when a disk group is successfully mounted, and automatically removes the disk group when it is dismounted (except for dismounts at instance shutdown).

ASM_DISKSTRING

Property	Description
Parameter type	String
Syntax	ASM_DISKSTRING = <i>discovery_string</i> [, <i>discovery_string</i>] ...
Default value	The null string; Automatic Storage Management discovery finds all disks in an operating system-specific location to which the Automatic Storage Management instance has read/write access.
Modifiable	ALTER SYSTEM
Real Application Clusters	Multiple instances can have different values. Different nodes might see the same disks under different names; however, each instance must be able to use its ASM_DISKSTRING to discover the same physical media as the other nodes in the cluster.

Note: This parameter may only be specified in an Automatic Storage Management instance.

ASM_DISKSTRING specifies an operating system-dependent value used by Automatic Storage Management to limit the set of disks considered for discovery. When a new disk is added to a disk group, each Automatic Storage Management instance that has the disk group mounted must be able to discover the new disk using the value of ASM_DISKSTRING.

In most cases, the default value will be sufficient. Using a more restrictive value may reduce the time required for Automatic Storage Management to perform discovery, and thus improve disk group mount time or the time for adding a disk to a disk group. It may be necessary to dynamically change ASM_DISKSTRING before adding a disk so that the new disk will be discovered.

An attempt to dynamically modify ASM_DISKSTRING will be rejected and the old value retained if the new value cannot be used to discover a disk that is in a disk group that is already mounted.

ASM_POWER_LIMIT

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	1 to 11
Real Application Clusters	Multiple instances can have different values.

Note: This parameter may only be specified in an Automatic Storage Management instance.

ASM_POWER_LIMIT specifies the maximum power on an Automatic Storage Management instance for disk rebalancing. The higher the limit, the faster rebalancing will complete. Lower values will take longer, but consume fewer processing and I/O resources.

If the POWER clause of a rebalance operation is not specified, then the default power will be the value of ASM_POWER_LIMIT.

AUDIT_FILE_DEST

Property	Description
Parameter type	String
Syntax	AUDIT_FILE_DEST = 'directory'
Default value	ORACLE_HOME/rdbms/audit
Modifiable	ALTER SYSTEM ... DEFERRED
Basic	No

AUDIT_FILE_DEST specifies the operating system directory into which the audit trail is written when the AUDIT_TRAIL initialization parameter is set to os. It is also the location to which mandatory auditing information is written and, if so specified by the AUDIT_SYS_OPERATIONS initialization parameter, audit records for user SYS.

AUDIT_SYS_OPERATIONS

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false
Basic	No

AUDIT_SYS_OPERATIONS enables or disables the auditing of operations issued by user SYS, and users connecting with SYSDBA or SYSOPER privileges. The audit records are written to the operating system's audit trail.

AUDIT_TRAIL

Property	Description
Parameter type	String
Syntax	AUDIT_TRAIL = { db os none true false db_extended }
Default value	none
Modifiable	No
Basic	No

AUDIT_TRAIL enables or disables database auditing.

Values:

- none or false
Disables database auditing.
- os
Enables database auditing and directs all audit records to the operating system's audit trail.
- db or true
Enables database auditing and directs all audit records to the database audit trail (the SYS.AUD\$ table).
- db_extended
Enables database auditing and directs all audit records to the database audit trail (the SYS.AUD\$ table). In addition, populates the SQLBIND and SQLTEXT CLOB columns of the SYS.AUD\$ table.

You can use the SQL statement AUDIT to set auditing options regardless of the setting of this parameter.

BACKGROUND_CORE_DUMP

Property	Description
Parameter type	String
Syntax	BACKGROUND_CORE_DUMP = { partial full }
Default value	partial
Modifiable	No
Basic	No

BACKGROUND_CORE_DUMP specifies whether Oracle includes the SGA in the core file for Oracle background processes.

Values:

- partial
Oracle does not include the SGA in the core dump.
- full
Oracle includes the SGA in the core dump.

See Also: ["SHADOW_CORE_DUMP"](#) on page 1-126

BACKGROUND_DUMP_DEST

Property	Description
Parameter type	String
Syntax	BACKGROUND_DUMP_DEST = { pathname directory }

Property	Description
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM
Range of values	Any valid local path, directory, or disk
Basic	No

BACKGROUND_DUMP_DEST specifies the pathname (directory or disc) where debugging trace files for the background processes (LGWR, DBW n , and so on) are written during Oracle operations.

An **alert file** in the directory specified by BACKGROUND_DUMP_DEST logs significant database events and messages. Anything that affects the database instance or global database is recorded here. The alert file is a normal text file. Its filename is operating system-dependent. For platforms that support multiple instances, it takes the form `alert_`*sid*`.log`, where *sid* is the system identifier. This file grows slowly, but without limit, so you might want to delete it periodically. You can delete the file even when the database is running.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter and on alert files
- Your operating system-specific Oracle documentation for the default value of this parameter
- "[USER_DUMP_DEST](#)" on page 1-144 for information on setting a destination for server process trace files

BACKUP_TAPE_IO_SLAVES

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM ... DEFERRED
Range of values	true false
Basic	No

BACKUP_TAPE_IO_SLAVES specifies whether I/O server processes (also called **slaves**) are used by Recovery Manager to back up, copy, or restore data to tape. When the value is set to `true`, Oracle uses an I/O server process to write to or read from a tape device. When the value is set to `false` (the default), Oracle does not use I/O server process for backups. Instead, the shadow process engaged in the backup accesses the tape device.

Note: You cannot perform duplexed backups unless you enable this parameter. Otherwise, Oracle returns an error. When this parameter is enabled, Recovery Manager will configure as many server processes as needed for the number of backup copies requested.

See Also:

- *Oracle Database Backup and Recovery Advanced User's Guide* for more information on duplexed backups
- "[DBWR_IO_SLAVES](#)" on page 1-44

BITMAP_MERGE_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	1048576 (1 MB)
Modifiable	No
Range of values	Operating system-dependent
Basic	No

Note: Oracle does not recommend using the `BITMAP_MERGE_AREA_SIZE` parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting `PGA_AGGREGATE_TARGET` instead. `BITMAP_MERGE_AREA_SIZE` is retained for backward compatibility.

`BITMAP_MERGE_AREA_SIZE` is relevant only for systems containing bitmap indexes. It specifies the amount of memory Oracle uses to merge bitmaps retrieved from a range scan of the index. The default value is 1 MB. A larger value usually improves performance, because the bitmap segments must be sorted before being merged into a single bitmap.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

BLANK_TRIMMING

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false
Basic	No

`BLANK_TRIMMING` specifies the data assignment semantics of character datatypes.

Values:

- `TRUE`
Allows the data assignment of a source character string or variable to a destination character column or variable even though the source length is longer than the

destination length. In this case, however, the additional length over the destination length is all blanks. This value complies with the semantics of SQL92 Transitional Level and above.

- FALSE

Disallows the data assignment if the source length is longer than the destination length and reverts to SQL92 Entry Level semantics.

CIRCUITS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

CIRCUITS specifies the total number of virtual circuits that are available for inbound and outbound network sessions. It is one of several parameters that contribute to the total SGA requirements of an instance.

You should not specify a value for this parameter unless you want to limit the number of virtual circuits.

See Also: *Oracle Database Concepts* for more information on memory structures and processes

CLUSTER_DATABASE

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

CLUSTER_DATABASE is a Real Application Clusters parameter that specifies whether or not Real Application Clusters is enabled.

See Also: *Oracle Real Application Clusters Administrator's Guide*

CLUSTER_DATABASE_INSTANCES

Property	Description
Parameter type	Integer

Property	Description
Default value	1
Modifiable	No
Range of values	Any nonzero value
Basic	No

CLUSTER_DATABASE_INSTANCES is a Real Application Clusters parameter that specifies the number of instances that are configured as part of your cluster database. You must set this parameter for every instance. Normally you should set this parameter to the number of instances in your Real Application Clusters environment. A proper setting for this parameter can improve memory use.

Oracle uses the value of this parameter to compute the default value of the LARGE_POOL_SIZE parameter when the PARALLEL_AUTOMATIC_TUNING parameter is set to true.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution
- *Oracle Real Application Clusters Administrator's Guide* for information on Real Application Clusters

CLUSTER_INTERCONNECTS

Property	Description
Parameter type	String
Syntax	CLUSTER_INTERCONNECTS = <i>ifn</i> [: <i>ifn</i>] ...
Default value	There is no default value.
Modifiable	No
Range of values	One or more IP addresses, separated by colons
Basic	No

CLUSTER_INTERCONNECTS provides Oracle with information about additional cluster interconnects available for use in Real Application Clusters environments.

The CLUSTER_INTERCONNECTS parameter can be used to override the default interconnect with a preferred cluster traffic network. This parameter is useful in Data Warehouse systems that have reduced availability requirements and high interconnect bandwidth demands. You can also use CLUSTER_INTERCONNECTS to override the default interconnect chosen by Oracle.

For example, if you are running two instances of Oracle for two databases on the same machine, then you can load balance the interconnect traffic to different physical interconnects. This does not reduce Oracle availability.

See Also: *Oracle Real Application Clusters Administrator's Guide*

COMMIT_POINT_STRENGTH

Property	Description
Parameter type	Integer
Default value	1
Modifiable	No
Range of values	0 to 255
Basic	No

COMMIT_POINT_STRENGTH is relevant only in distributed database systems. It specifies a value that determines the **commit point site** in a distributed transaction. The node in the transaction with the highest value for COMMIT_POINT_STRENGTH will be the commit point site.

The commit point site of a database should reflect the amount of critical shared data in the database. For example, a database on a mainframe computer typically shares more data among users than one on a personal computer. Therefore, COMMIT_POINT_STRENGTH should be set to a higher value for the mainframe computer.

The commit point site stores information about the status of transactions. Other computers in a distributed transaction require this information during Oracle's two-phase commit, so it is desirable to have machines that are always available as commit point sites. Therefore, set COMMIT_POINT_STRENGTH to a higher value on your more available machines.

See Also:

- *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for information on two-phase commit
- Your operating system-specific Oracle documentation for the default value of this parameter

COMPATIBLE

Property	Description
Parameter type	String
Syntax	COMPATIBLE = <i>release_number</i>
Default value	10.0.0
Modifiable	No
Range of values	9.2.0 to default release
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

COMPATIBLE allows you to use a new release of Oracle, while at the same time guaranteeing backward compatibility with an earlier release. This is helpful if it becomes necessary to revert to the earlier release.

This parameter specifies the release with which Oracle must maintain compatibility. It allows you to take advantage of the maintenance improvements of a new release immediately in your production systems without testing the new functionality in your environment. Some features of the release may be restricted.

When using a standby database, this parameter must have the same value on both the primary and standby databases.

See Also: *Oracle Database Upgrade Guide* for more information on setting this parameter

CONTROL_FILE_RECORD_KEEP_TIME

Property	Description
Parameter type	Integer
Default value	7 (days)
Modifiable	ALTER SYSTEM
Range of values	0 to 365 (days)
Basic	No

CONTROL_FILE_RECORD_KEEP_TIME specifies the minimum number of days before a reusable record in the control file can be reused. In the event a new record needs to be added to a reusable section and the oldest record has not aged enough, the record section expands. If this parameter is set to 0, then reusable sections never expand, and records are reused as needed.

Note: This parameter applies only to records in the control file that are circularly reusable (such as archive log records and various backup records). It does not apply to records such as datafile, tablespace, and redo thread records, which are never reused unless the corresponding object is dropped from the tablespace.

See Also:

- ["CONTROL_FILES"](#) on page 1-28
- *Oracle Database Backup and Recovery Basics*

CONTROL_FILES

Property	Description
Parameter type	String
Syntax	CONTROL_FILES = <i>filename</i> [, <i>filename</i>] ... Note: The control file name can be an OMF (Oracle Managed Files) name. This occurs when the control file is re-created using the CREATE CONTROLFILE REUSE statement.
Default value	Operating system-dependent
Modifiable	No

Property	Description
Range of values	1 to 8 filenames
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

Every database has a **control file**, which contains entries that describe the structure of the database (such as its name, the timestamp of its creation, and the names and locations of its datafiles and redo files). `CONTROL_FILES` specifies one or more names of control files, separated by commas.

Oracle recommends that you multiplex multiple control files on different devices or mirror the file at the operating system level.

See Also:

- *Oracle Database Performance Tuning Guide*
- *Oracle Database Administrator's Guide*

CORE_DUMP_DEST

Property	Description
Parameter type	String
Syntax	<code>CORE_DUMP_DEST = <i>directory</i></code>
Default value	<code>ORACLE_HOME/DBS</code>
Modifiable	<code>ALTER SYSTEM</code>
Basic	No

`CORE_DUMP_DEST` is primarily a UNIX parameter and may not be supported on your platform. It specifies the directory where Oracle dumps core files.

CPU_COUNT

Property	Description
Parameter type	Integer
Default value	Set automatically by Oracle
Modifiable	No
Range of values	0 to unlimited
Basic	No

Caution: On most platforms, Oracle automatically sets the value of `CPU_COUNT` to the number of CPUs available to your Oracle instance. Do not change the value of `CPU_COUNT`.

CPU_COUNT specifies the number of CPUs available to Oracle. On single-CPU computers, the value of CPU_COUNT is 1.

See Also: Your operating system-specific Oracle documentation for more information about this parameter

CREATE_BITMAP_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	8388608 (8 MB)
Modifiable	No
Range of values	Operating system-dependent
Basic	No

Note: Oracle does not recommend using the CREATE_BITMAP_AREA_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. CREATE_BITMAP_AREA_SIZE is retained for backward compatibility.

CREATE_BITMAP_AREA_SIZE is relevant only for systems containing bitmap indexes. It specifies the amount of memory (in bytes) allocated for bitmap creation. The default value is 8 MB. A larger value may speed up index creation.

Cardinality is the number of unique values in a column in relation to the number of rows in the table. If cardinality is very small, you can set a small value for this parameter. For example, if cardinality is only 2, then the value can be on the order of kilobytes rather than megabytes. As a general rule, the higher the cardinality, the more memory is needed for optimal performance.

See Also: *Oracle Database Performance Tuning Guide* for more information on using bitmap indexes

CREATE_STORED_OUTLINES

Property	Description
Parameter type	String
Syntax	CREATE_STORED_OUTLINES = { true false <i>category_name</i> }
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

CREATE_STORED_OUTLINES determines whether Oracle automatically creates and stores an outline for each query submitted during the session.

Values:

- `true`
Enables automatic outline creation for subsequent queries in the same session. These outlines receive a unique system-generated name and are stored in the `DEFAULT` category. If a particular query already has an outline defined for it in the `DEFAULT` category, then that outline will remain and a new outline will not be created.
- `false`
Disables automatic outline creation during the session. This is the default.
- `category_name`
Enables the same behavior as `true` except that any outline created during the session is stored in the `category_name` category.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

CURSOR_SHARING

Property	Description
Parameter type	String
Syntax	<code>CURSOR_SHARING = { SIMILAR EXACT FORCE }</code>
Default value	<code>EXACT</code>
Modifiable	<code>ALTER SESSION, ALTER SYSTEM</code>
Basic	No

`CURSOR_SHARING` determines what kind of SQL statements can share the same cursors.

Values:

- `FORCE`
Forces statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect the meaning of the statement.
- `SIMILAR`
Causes statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect either the meaning of the statement or the degree to which the plan is optimized.
- `EXACT`
Only allows statements with identical text to share the same cursor.

Note: Forcing cursor sharing among similar (but not identical) statements can have unexpected results in some DSS applications, or applications that use stored outlines.

See Also: *Oracle Database Performance Tuning Guide* for information on setting this parameter in these and other environments

CURSOR_SPACE_FOR_TIME

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false
Basic	No

`CURSOR_SPACE_FOR_TIME` lets you use more space for cursors in order to save time. It affects both the shared SQL area and the client's private SQL area.

Values:

- `TRUE`

Shared SQL areas are kept pinned in the shared pool. As a result, shared SQL areas are not aged out of the pool as long as an open cursor references them. Because each active cursor's SQL area is present in memory, execution is faster. However, the shared SQL areas never leave memory while they are in use. Therefore, you should set this parameter to `TRUE` only when the shared pool is large enough to hold all open cursors simultaneously.

In addition, a setting of `TRUE` retains the private SQL area allocated for each cursor between executions instead of discarding it after cursor execution, saving cursor allocation and initialization time

- `FALSE`

Shared SQL areas can be deallocated from the library cache to make room for new SQL statements.

Note: If this parameter is set to `TRUE`, the `SERIAL_REUSE` parameter is disabled.

See Also:

- *Oracle Database Concepts* and *Oracle Database Performance Tuning Guide*
- ["SERIAL_REUSE"](#) on page 1-122

DB_nK_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_[2 4 8 16 32]K_CACHE_SIZE = integer [K M G]</code>
Default value	0 (additional block size caches are not configured by default)
Modifiable	<code>ALTER SYSTEM</code>

Property	Description
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

DB_ *n*K_CACHE_SIZE (where *n* = 2, 4, 8, 16, 32) specifies the size of the cache for the *n*K buffers. You can set this parameter only when DB_BLOCK_SIZE has a value other than *n*K. For example, if DB_BLOCK_SIZE=4096, then it is illegal to specify the parameter DB_4K_CACHE_SIZE (because the size for the 4 KB block cache is already specified by DB_CACHE_SIZE).

Do not set this parameter to zero if there are any online tablespaces with an *n*K block size.

Operating system-specific block size restrictions apply. For example, you cannot set DB_32K_CACHE_SIZE if the operating system's maximum block size is less than 32 KB. Also, you cannot set DB_2K_CACHE_SIZE if the minimum block size is greater than 2 KB.

See Also: Your operating system-specific Oracle documentation for more information on block size restrictions

DB_BLOCK_BUFFERS

Property	Description
Parameter type	Integer
Default value	Derived: 48 MB / DB_BLOCK_SIZE
Modifiable	No
Range of values	50 to an operating system-specific maximum
Basic	No
Real Application Clusters	Multiple instances can have different values, and you can change the values as needed.

Note: DB_BLOCK_BUFFERS cannot be combined with the dynamic DB_CACHE_SIZE parameter; combining these parameters in the same parameter file will produce an error. DB_BLOCK_BUFFERS is retained for backward compatibility.

DB_BLOCK_BUFFERS specifies the number of database buffers in the buffer cache. It is one of several parameters that contribute to the total memory requirements of the SGA of an instance.

This parameter, together with DB_BLOCK_SIZE, determines the total size of the buffer cache. Effective use of the buffer cache can greatly reduce the I/O load on the database. DB_BLOCK_SIZE can be specified only when the database is first created, so you use DB_BLOCK_BUFFERS to control the size of the buffer cache.

DB_BLOCK_CHECKING

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false
Basic	No

DB_BLOCK_CHECKING controls whether Oracle performs block checking for data blocks. When this parameter is set to `true`, Oracle performs block checking for all data blocks. When it is set to `false`, Oracle does not perform block checking for blocks in the user tablespaces. However, block checking for the SYSTEM tablespace is always turned on.

Oracle checks a block by going through the data on the block, making sure it is self-consistent. Block checking can often prevent memory and data corruption. Block checking typically causes 1% to 10% overhead, depending on workload. The more updates or inserts in a workload, the more expensive it is to turn on block checking. You should set DB_BLOCK_CHECKING to `true` if the performance overhead is acceptable.

DB_BLOCK_CHECKSUM

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

DB_BLOCK_CHECKSUM determines whether DBWn and the direct loader will calculate a **checksum** (a number calculated from all the bytes stored in the block) and store it in the cache header of every data block when writing it to disk. Checksums are verified when a block is read-only if this parameter is `true` and the last write of the block stored a checksum. In addition, Oracle gives every log block a checksum before writing it to the current log.

If this parameter is set to `false`, DBWn calculates checksums only for the SYSTEM tablespace, but not for user tablespaces.

Checksums allow Oracle to detect corruption caused by underlying disks, storage systems, or I/O systems. Turning on this feature typically causes only an additional 1% to 2% overhead. Therefore, Oracle recommends that you set DB_BLOCK_CHECKSUM to `true`.

DB_BLOCK_SIZE

Property	Description
Parameter type	Integer
Default value	8192
Modifiable	No
Range of values	2048 to 32768, but your operating system may have a narrower range
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

Caution: Set this parameter at the time of database creation. Do not alter it afterward.

DB_BLOCK_SIZE specifies (in bytes) the size of Oracle database blocks. Typical values are 4096 and 8192. The value of this parameter must be a multiple of the physical block size at the device level.

The value for DB_BLOCK_SIZE in effect at the time you create the database determines the size of the blocks. The value must remain set to its initial value.

For Real Application Clusters, this parameter affects the maximum value of the FREELISTS storage parameter for tables and indexes. Oracle uses one database block for each freelist group. Decision support system (DSS) and data warehouse database environments tend to benefit from larger block size values.

See Also:

- *Oracle Database Performance Tuning Guide* and *Oracle Database Administrator's Guide* for information on setting this parameter
- *Oracle Database SQL Reference* for information on freelist groups
- *Oracle Database Performance Tuning Guide* for more information on the DSS and data warehouse environments
- *Oracle Database Concepts* for general information on block sizes

DB_CACHE_ADVICE

Property	Description
Parameter type	String
Syntax	DB_CACHE_ADVICE = { ON READY OFF }
Default value	If STATISTICS_LEVEL is set to TYPICAL or ALL, then ON If STATISTICS_LEVEL is set to BASIC, then OFF
Modifiable	ALTER SYSTEM
Basic	No

DB_CACHE_ADVICE enables or disables statistics gathering used for predicting behavior with different cache sizes through the V\$DB_CACHE_ADVICE performance view.

Values:

- OFF

Advisory is turned off and the memory for the advisory is not allocated.

- READY

Advisory is turned off but the memory for the advisory remains allocated. Allocating the memory before the advisory is actually turned on avoids the risk of an error when you switch the parameter to ON.

If the parameter is switched to this state from ON, the contents of the view are preserved and the memory for the advisory is retained.

If the parameter is switched to this state from OFF, you may get an error.

- ON

Advisory is turned on. CPU and memory overheads are incurred. Attempting to set the parameter to this state when it is already in the OFF state may result in an error. Otherwise, the view (V\$DB_CACHE_ADVICE) is reset and statistics are gathered to the newly refreshed view.

If the parameter is in the READY state, you can set it to ON without any errors because the memory is already allocated. The view is reset and statistics are displayed in the newly refreshed view.

DB_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	DB_CACHE_SIZE = <i>integer</i> [K M G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set: 48 MB, rounded up to the nearest granule size
Modifiable	ALTER SYSTEM
Basic	No

DB_CACHE_SIZE specifies the size of the DEFAULT buffer pool for buffers with the primary block size (the block size defined by the DB_BLOCK_SIZE initialization parameter).

The value must be at least the size of one granule (smaller values are automatically rounded up to the granule size). A value of zero is illegal because zero is the size of the DEFAULT pool for the standard block size, which is the block size for the SYSTEM tablespace.

See Also: *Oracle Database Performance Tuning Guide* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DB_CREATE_FILE_DEST

Property	Description
Parameter type	String
Syntax	DB_CREATE_FILE_DEST = <i>directory</i> <i>disk group</i>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	Yes

DB_CREATE_FILE_DEST specifies the default location for Oracle-managed datafiles. This location is also used as the default location for Oracle-managed control files and online redo logs if none of the DB_CREATE_ONLINE_LOG_DEST_n initialization parameters are specified.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle-managed file.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter and on Oracle-managed files

DB_CREATE_ONLINE_LOG_DEST_n

Property	Description
Parameter type	String
Syntax	DB_CREATE_ONLINE_LOG_DEST_[1 2 3 4 5] = <i>directory</i> <i>disk group</i>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	Yes

DB_CREATE_ONLINE_LOG_DEST_n (where $n = 1, 2, 3, \dots, 5$) specifies the default location for Oracle-managed control files and online redo logs. If more than one DB_CREATE_ONLINE_LOG_DEST_n parameter is specified, then the control file or online redo log is multiplexed across the locations of the other DB_CREATE_ONLINE_LOG_DEST_n parameters. One member of each online redo log is created in each location, and one control file is created in each location.

Specifying at least two parameters provides greater fault tolerance for the control files and online redo logs if one of the locations should fail.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle-managed file.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter and on Oracle-managed files

DB_DOMAIN

Property	Description
Parameter type	String
Syntax	DB_DOMAIN = <i>domain_name</i>
Default value	There is no default value.
Modifiable	No
Range of values	Any legal string of name components, separated by periods and up to 128 characters long (including the periods). This value cannot be NULL.
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

In a distributed database system, DB_DOMAIN specifies the logical location of the database within the network structure. You should set this parameter if this database is or ever will be part of a distributed system. The value consists of the extension components of a global database name, consisting of valid identifiers, separated by periods. Oracle recommends that you specify DB_DOMAIN as a unique string for all databases in a domain.

This parameter allows one department to create a database without worrying that it might have the same name as a database created by another department. If one sales department's DB_DOMAIN is JAPAN.ACME.COM, then their SALES database (SALES.JAPAN.ACME.COM) is uniquely distinguished from another database with DB_NAME = SALES but with DB_DOMAIN = US.ACME.COM.

If you omit the domains from the name of a database link, Oracle expands the name by qualifying the database with the domain of your local database as it currently exists in the data dictionary, and then stores the link name in the data dictionary. The characters valid in a database domain name are: alphanumeric characters, underscore (_), and number sign (#).

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- The data dictionary view "[GLOBAL_NAME](#)" on page 3-138

DB_FILE_MULTIBLOCK_READ_COUNT

Property	Description
Parameter type	Integer
Default value	8
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	Operating system-dependent
Basic	No

DB_FILE_MULTIBLOCK_READ_COUNT is one of the parameters you can use to minimize I/O during table scans. It specifies the maximum number of blocks read in one I/O operation during a sequential scan. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multiblock read count, and whether parallel execution is being utilized for the operation.

Online transaction processing (OLTP) and batch environments typically have values in the range of 4 to 16 for this parameter. DSS and data warehouse environments tend to benefit most from maximizing the value of this parameter. The optimizer is more likely to choose a full table scan over an index if the value of this parameter is high.

The maximum value is always less than the operating system's maximum I/O size expressed as Oracle blocks ((max I/O size)/DB_BLOCK_SIZE). If you set this parameter to a value greater than the maximum, Oracle uses the maximum.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value of this parameter

DB_FILE_NAME_CONVERT

Property	Description
Parameter type	String
Syntax	<p>DB_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' , ...</p> <p>Where:</p> <ul style="list-style-type: none"> ■ <i>string1</i> is the pattern of the primary database filename ■ <i>string2</i> is the pattern of the standby database filename ■ <i>string3</i> is the pattern of the primary database filename ■ <i>string4</i> is the pattern of the standby database filename <p>You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks.</p> <p>The following are example settings that are acceptable:</p> <pre>DB_FILE_NAME_CONVERT=' /dbs/t1/' , '/dbs/t1/s_' , 'dbs/t2/' , 'dbs/t2/s_'</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION
Basic	No

DB_FILE_NAME_CONVERT is useful for creating a duplicate database for recovery purposes. It converts the filename of a new datafile on the primary database to a filename on the standby database. If you add a datafile to the primary database, you must add a corresponding file to the standby database. When the standby database is updated, this parameter converts the datafile name on the primary database to the datafile name on the standby database. The file on the standby database must exist and be writable, or the recovery process will halt with an error.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is signalled during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalued parameters).

Set the value of this parameter to two strings. The first string is the pattern found in the datafile names on the primary database. The second string is the pattern found in the datafile names on the standby database.

You can also use `DB_FILE_NAME_CONVERT` to rename the datafiles in the clone controlfile when setting up a clone database during tablespace point-in-time recovery.

See Also:

- *Oracle Database Backup and Recovery Advanced User's Guide*
- *Oracle Data Guard Concepts and Administration*

DB_FILES

Property	Description
Parameter type	Integer
Default value	200
Modifiable	No
Range of values	Minimum: the largest among the absolute file numbers of the datafiles in the database Maximum: operating system-dependent
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

`DB_FILES` specifies the maximum number of database files that can be opened for this database. The maximum valid value is the maximum number of files, subject to operating system constraint, that will ever be specified for the database, including files to be added by `ADD DATAFILE` statements.

If you increase the value of `DB_FILES`, then you must shut down and restart all instances accessing the database before the new value can take effect. If you have a primary and standby database, then they should have the same value for this parameter.

See Also:

- *Oracle Real Application Clusters Administrator's Guide* for information on setting this parameter in a Real Application Clusters environment
- Your operating system-specific Oracle documentation for the default value of this parameter

DB_FLASHBACK_RETENTION_TARGET

Property	Description
Parameter type	Integer
Default value	1440 (minutes)
Modifiable	ALTER SYSTEM
Range of values	0 to $2^{32} - 1$ (max value represented by 32 bits)
Basic	No

DB_FLASHBACK_RETENTION_TARGET specifies the upper limit (in minutes) on how far back in time the database may be flashed back. How far back one can flashback a database depends on how much flashback data Oracle has kept in the flash recovery area.

DB_KEEP_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	DB_KEEP_CACHE_SIZE = <i>integer</i> [K M G]
Default value	0 (KEEP cache is not configured by default)
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

DB_KEEP_CACHE_SIZE specifies the size of the KEEP buffer pool. The size of the buffers in the KEEP buffer pool is the primary block size (the block size defined by the DB_BLOCK_SIZE initialization parameter).

See Also:

- "[DB_RECYCLE_CACHE_SIZE](#)" on page 1-43
- *Oracle Database Performance Tuning Guide* for information on setting these parameters and on using multiple buffer pools

DB_NAME

Property	Description
Parameter type	String
Syntax	DB_NAME = <i>database_name</i>
Default value	There is no default value.
Modifiable	No
Basic	Yes

Property	Description
Real Application Clusters	You must set this parameter for every instance. Multiple instances must have the same value, or the same value must be specified in the <code>STARTUP OPEN SQL*Plus</code> command or the <code>ALTER DATABASE MOUNT SQL</code> statement.

`DB_NAME` specifies a database identifier of up to 8 characters. This parameter must be specified and must correspond to the name specified in the `CREATE DATABASE` statement.

If you have multiple databases, the value of this parameter should match the Oracle instance identifier of each one to avoid confusion with other databases running on the system. The value of `DB_NAME` should be the same in both the standby and production initialization parameter files.

The database name specified in either the `STARTUP` command or the `ALTER DATABASE . . . MOUNT` statement for each instance of the cluster database must correspond to the `DB_NAME` initialization parameter setting.

The following characters are valid in a database name: alphanumeric characters, underscore (`_`), number sign (`#`), and dollar sign (`$`). No other characters are valid. Oracle removes double quotation marks before processing the database name. Therefore you cannot use double quotation marks to embed other characters in the name. The database name is case insensitive.

See Also: *Oracle Database Administrator's Guide* and *Oracle Real Application Clusters Administrator's Guide* for more information on setting this parameter

DB_RECOVERY_FILE_DEST

Property	Description
Parameter type	String
Syntax	<code>DB_RECOVERY_FILE_DEST = directory disk group</code>
Default value	There is no default value.
Modifiable	<code>ALTER SYSTEM . . . SID='*'</code>
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

`DB_RECOVERY_FILE_DEST` specifies the default location for the flash recovery area. The flash recovery area contains multiplexed copies of current control files and online redo logs, as well as archived redo logs, flashback logs, and RMAN backups.

Specifying this parameter without also specifying the `DB_RECOVERY_FILE_DEST_SIZE` initialization parameter is not allowed.

See Also:

- *Oracle Database Backup and Recovery Basics* for information on setting up and configuring the flash recovery area
- ["DB_RECOVERY_FILE_DEST_SIZE"](#) on page 1-43

DB_RECOVERY_FILE_DEST_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_RECOVERY_FILE_DEST_SIZE = integer [K M G]</code>
Default value	There is no default value.
Modifiable	<code>ALTER SYSTEM ... SID='*'</code>
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

`DB_RECOVERY_FILE_DEST_SIZE` specifies (in bytes) the hard limit on the total space to be used by target database recovery files created in the flash recovery area.

See Also:

- *Oracle Database Backup and Recovery Basics* for information on setting up and configuring the flash recovery area
- "[DB_RECOVERY_FILE_DEST](#)" on page 1-42

DB_RECYCLE_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_RECYCLE_CACHE_SIZE = integer [K M G]</code>
Default value	0 (RECYCLE cache is not configured by default)
Modifiable	<code>ALTER SYSTEM</code>
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

`DB_RECYCLE_CACHE_SIZE` specifies the size of the RECYCLE buffer pool. The size of the buffers in the RECYCLE pool is the primary block size (the block size defined by the `DB_BLOCK_SIZE` initialization parameter).

See Also:

- "[DB_KEEP_CACHE_SIZE](#)" on page 1-41
- *Oracle Database Performance Tuning Guide* for information on setting these parameters and on using multiple buffer pools

DB_UNIQUE_NAME

Property	Description
Parameter type	String

Property	Description
Syntax	DB_UNIQUE_NAME = <i>database_unique_name</i>
Default value	Database instances: the value of DB_NAME Automatic Storage Management instances: +ASM
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

DB_UNIQUE_NAME specifies a globally unique name for the database. Databases with the same DB_NAME within the same DB_DOMAIN (for example, copies of a database created for reporting or a physical standby) must have a unique DB_UNIQUE_NAME. Every database's DB_UNIQUE_NAME must be unique within the enterprise.

The value of DB_UNIQUE_NAME can be up to 30 characters and is case insensitive. The following characters are valid in a database name: alphanumeric characters, underscore (_), number sign (#), and dollar sign (\$).

See Also: *Oracle Data Guard Concepts and Administration* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DB_WRITER_PROCESSES

Property	Description
Parameter type	Integer
Default value	1 or CPU_COUNT / 8, whichever is greater
Modifiable	No
Range of values	1 to 20
Basic	No

DB_WRITER_PROCESSES is useful for systems that modify data heavily. It specifies the initial number of database writer processes for an instance.

See Also:

- *Oracle Database Concepts* for more information on the database writer processes
- *Oracle Database Performance Tuning Guide* for information on setting the DBWn parameters

DBWR_IO_SLAVES

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No

Property	Description
Range of values	0 to operating system-dependent
Basic	No

DBWR_IO_SLAVES is relevant only on systems with only one database writer process (DBW0). It specifies the number of I/O server processes used by the DBW0 process. The DBW0 process and its server processes always write to disk. By default, the value is 0 and I/O server processes are not used.

If you set DBWR_IO_SLAVES to a nonzero value, the number of I/O server processes used by the ARCH and LGWR processes is set to 4. However, the number of I/O server processes used by Recovery Manager is set to 4 only if asynchronous I/O is disabled (either your platform does not support asynchronous I/O or `disk_asynch_io` is set to false).

Typically, I/O server processes are used to simulate asynchronous I/O on platforms that do not support asynchronous I/O or that implement it inefficiently. However, you can use I/O server processes even when asynchronous I/O is being used. In that case the I/O server processes will use asynchronous I/O.

I/O server processes are also useful in database environments with very large I/O throughput, even if asynchronous I/O is enabled.

See Also: ["BACKUP_TAPE_IO_SLAVES"](#) on page 1-23

DDL_WAIT_FOR_LOCKS

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false
Basic	No

DDL_WAIT_FOR_LOCKS specifies whether DDL statements (such as `ALTER TABLE ... ADD COLUMN`) wait and complete instead of timing out if the statement is not able to acquire all required locks.

Values:

- true
DDL statements wait until the statement acquires all required locks
- false
DDL statements time out if the statement cannot obtain all required locks

DG_BROKER_CONFIG_FILEn

Property	Description
Parameter type	String

Property	Description
Syntax	DG_BROKER_CONFIG_FILE[1 2] = <i>filename</i>
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM
Range of values	One filename
Basic	No

DG_BROKER_CONFIG_FILE n (where $n = 1, 2$) specifies the names for the Data Guard broker configuration files.

Every database that is part of a Data Guard broker configuration has two broker configuration files, which contain entries that describe the state and properties of the configuration (such as the sites and databases that are part of the configuration, the roles and properties of each of the databases, and the state of each of the elements of the configuration). Two files are provided so as to always maintain the last known good state of the configuration.

If DG_BROKER_CONFIG_FILE n is not explicitly defined, then it is set to an operating system-specific default value at instance startup. The parameter can only be altered when the DMON (Data Guard broker) process is not running.

See Also: *Oracle Data Guard Broker* for more information about setting this parameter

DG_BROKER_START

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

DG_BROKER_START enables Oracle to determine whether or not the DMON (Data Guard broker) process should be started. DMON is a non-fatal Oracle background process and exists as long as the instance exists, whenever this parameter is set to true.

If the site is never going to be configured in a Data Guard broker configuration, then you can leave the parameter unspecified and accept the default value of false. If the site is part of a Data Guard broker configuration, then automatic startup of the DRMON process is simplified by setting DG_BROKER_START to true in the initialization parameter file.

DISK_ASYNC_IO

Property	Description
Parameter type	Boolean

Property	Description
Default value	true
Modifiable	No
Range of values	true false
Basic	No

DISK_ASYNC_IO controls whether I/O to datafiles, control files, and logfiles is asynchronous (that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans). If your platform supports asynchronous I/O to disk, Oracle recommends that you leave this parameter set to its default value. However, if the asynchronous I/O implementation is not stable, you can set this parameter to `false` to disable asynchronous I/O. If your platform does not support asynchronous I/O to disk, this parameter has no effect.

If you set `DISK_ASYNC_IO` to `false`, then you should also set `DBWR_IO_SLAVES` to a value other than its default of zero in order to simulate asynchronous I/O.

See Also:

- ["DBWR_IO_SLAVES"](#) on page 1-44
- *Oracle Database Performance Tuning Guide* for more information on asynchronous I/O and on setting this parameter

DISPATCHERS

Property	Description
Parameter type	String
Syntax	<pre>DISPATCHERS = 'dispatch_clause'</pre> <p>dispatch_clause::=</p> <pre>(PROTOCOL = protocol) (AADDRESS = address) (DESCRIPTION = description) [options_clause]</pre> <p>options_clause::=</p> <pre>(DISPATCHERS = integer SESSIONS = integer CONNECTIONS = integer TICKS = seconds POOL = {1 ON YES TRUE BOTH ({IN OUT} = ticks) 0 OFF NO FALSE ticks} MULTIPLEX = {1 ON YES TRUE 0 OFF NO FALSE BOTH IN OUT} LISTENER = tnsname SERVICE = service INDEX = integer)</pre>

Property	Description
Default value	If SHARED_SERVERS is greater than 0, then DISPATCHERS defaults to '(PROTOCOL=tcp)', causing 1 TCP/IP dispatcher to be created.
Modifiable	ALTER SYSTEM
Basic	No

DISPATCHERS configures dispatcher processes in the shared server architecture. The parsing software supports a name-value syntax to enable the specification of attributes in a position-independent, case-insensitive manner. For example:

```
DISPATCHERS = '(PROTOCOL=TCP)(DISPATCHERS=3)'
```

Attributes may be specified using the full attribute name or any substring beginning with the first 3 characters. For example, SESSIONS can be specified as SES, SESS, SESSI, and so on.

Specify only one of the following attributes: PROTOCOL, ADDRESS, or DESCRIPTION. If you specify either ADDRESS or DESCRIPTION, then you can specify additional network attributes. Doing so supports multi-homed hosts.

dispatch_clause

- PROTOCOL
The network protocol for which the dispatcher generates a listening endpoint.
- ADDRESS
The network protocol address of the endpoint on which the dispatchers listen.
- DESCRIPTION
The network description of the endpoint on which the dispatchers listen, including the protocol address.

options_clause

- DISPATCHERS
The initial number of dispatchers to start. The default is 1.
- SESSIONS
The maximum number of network sessions to allow for each dispatcher. The default is operating system-specific. Most operating systems have a default of 16 KB.
- CONNECTIONS
The maximum number of network connections to allow for each dispatcher. The default is operating system-specific.
- TICKS
The length of a network tick in seconds. The default is 1 second.
- POOL
Enables Connection Pooling.
 - An integer indicates that Connection Pooling is enabled for both incoming and outgoing network connections. The number specified is the timeout in ticks for both incoming and outgoing network connections.

- ON, YES, TRUE, and BOTH indicate that Connection Pooling is enabled for both incoming and outgoing network connections. A default timeout of 10 ticks is used for both incoming and outgoing network connections.
- IN indicates that Connection Pooling is enabled for incoming network connections and the default timeout of 10 ticks is used for incoming network connections.
- OUT indicates that Connection Pooling is enabled for outgoing network connections and the default timeout of 10 ticks is used for outgoing network connections.
- NO, OFF, and FALSE indicate that Connection Pooling is disabled for both incoming and outgoing network connections. This is the default.

POOL can also be assigned a name-value string such as "(IN=10)", "(OUT=20)", or "(IN=10)(OUT=20)". In such cases:

If an IN numeric value is specified, then Connection Pooling is enabled for incoming connections, and the number specified is the timeout in ticks for incoming network connections. If an OUT numeric value is specified, then Connection Pooling is enabled for outgoing network connections, and the number specified is the timeout in ticks for outgoing network connections. If the numeric value of a specified timeout is 1, then the default value of 10 ticks is used. If the numeric value is 0, then pooling is not on.

- MULTIPLEX

Enables the Oracle Connection Manager session multiplexing feature.

- The values 1, ON, YES, TRUE, and BOTH indicate that Network Session Multiplex is enabled for both incoming and outgoing network connections.
- The value IN indicates that Network Session Multiplex is enabled for incoming network connections.
- The value OUT indicates that Network Session Multiplexing is enabled for outgoing network connections.
- The values 0, NO, OFF, and FALSE indicate that Network Session Multiplexing is disabled for both incoming and outgoing network connections. This is the default.

- LISTENER

Specifies the network name of an address or address list of the Oracle Net listeners with which the dispatchers will register.

The LISTENER attribute facilitates administration of multi-homed hosts. This attribute specifies the appropriate listeners with which the dispatchers will register. The LISTENER attribute takes precedence over the LOCAL_LISTENER and REMOTE_LISTENER parameters. See "[LOCAL_LISTENER](#)" on page 1-68 and "[REMOTE_LISTENER](#)" on page 1-117.

- SERVICE

Specifies one or more names by which clients can connect to the dispatchers. The SERVICE attribute takes precedence over the SERVICE_NAMES parameter.

- INDEX

Use this attribute in an ALTER SYSTEM SET DISPATCHERS statement to indicate which dispatcher configuration you want to modify. (If you specify INDEX in the initialization parameter file, the Oracle Database ignores it.) In an ALTER SYSTEM statement, INDEX specifies the order in which the parameter's values were

initialized. The value ranges from 0 (for the first dispatcher configuration) to one less than the total number of dispatcher configurations you define.

For example, if you specify 3 dispatcher configurations in the initialization parameter file, you would modify the third dispatcher configuration by specifying `INDEX=2` in the `ALTER SYSTEM` statement. You could also add an additional dispatcher configuration in the `ALTER SYSTEM` statement by specifying `INDEX=3`.

If `INDEX` is not specified in the `ALTER SYSTEM` statement, then the `PROTOCOL`, `ADDRESS`, or `DESCRIPTION` attributes must be specified, and if a dispatcher configuration matching this `PROTOCOL`, `ADDRESS`, or `DESCRIPTION` exists, then that configuration will be modified. Otherwise, a new configuration will be added.

See Also: *Oracle Net Services Administrator's Guide* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DISTRIBUTED_LOCK_TIMEOUT

Property	Description
Parameter type	Integer
Default value	60
Modifiable	No
Range of values	1 to unlimited
Basic	No

`DISTRIBUTED_LOCK_TIMEOUT` specifies the amount of time (in seconds) for distributed transactions to wait for locked resources.

See Also: *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for more information on data concurrency

DML_LOCKS

Property	Description
Parameter type	Integer
Default value	Derived: 4 * <code>TRANSACTIONS</code>
Modifiable	No
Range of values	20 to unlimited; a setting of 0 disables enqueues
Basic	No
Real Application Clusters	You must set this parameter for every instance, and all instances must have positive values or all must be 0.

A **DML lock** is a lock obtained on a table that is undergoing a DML operation (insert, update, delete). `DML_LOCKS` specifies the maximum number of DML locks—one for each table modified in a transaction. The value should equal the grand total of locks on tables currently referenced by all users. For example, if three users are modifying data in one table, then three entries would be required. If three users are modifying data in two tables, then six entries would be required.

The default value assumes an average of four tables referenced for each transaction. For some systems, this value may not be enough.

Enqueues are shared memory structures that serialize access to database resources. If you set the value of `DML_LOCKS` to 0, enqueues are disabled and performance is slightly increased. However, you cannot use `DROP TABLE`, `CREATE INDEX`, or explicit lock statements such as `LOCK TABLE IN EXCLUSIVE MODE`.

Oracle holds more locks during parallel DML than during serial execution. Therefore, if your database supports a lot of parallel DML, you may need to increase the value of this parameter.

See Also:

- *Oracle Database Concepts* for a discussion of lock and enqueue resources needed for parallel DML
- *Oracle Real Application Clusters Administrator's Guide*, *Oracle Database Concepts*, and *Oracle Database Administrator's Guide* for more information on data concurrency

ENQUEUE_RESOURCES

Property	Description
Parameter type	Integer
Default value	Derived from <code>SESSIONS</code> parameter
Modifiable	No
Range of values	10 to unlimited
Basic	No

`ENQUEUE_RESOURCES` sets the number of resources that can be concurrently locked by the lock manager. An **enqueue** is a sophisticated locking mechanism that permits several concurrent processes to share known resources to varying degrees. Any object that can be used concurrently can be protected with enqueues. For example, Oracle allows varying levels of sharing on tables: two processes can lock a table in share mode or in share update mode.

One difference between enqueues and latches is that latches do not entail an ordered queue of waiting processes as do enqueues. Processes waiting for latches can either use timers to wake up and retry or (in multiprocessors) spin.

At database startup time, Oracle allocates the number of enqueues specified by the `ENQUEUE_RESOURCES` parameter. The default value of `ENQUEUE_RESOURCES` is derived from the `SESSIONS` parameter and is usually adequate, as long as its value is greater than `DML_LOCKS + 20`. For three or fewer sessions, the default value is the number of database files + 20. For 4 to 10 sessions, the default value is the number of database files + $((SESSIONS - 3) * 5) + 20$. For more than 10 sessions, it is the number of database files + $((SESSIONS - 10) * 2) + 55$.

If you explicitly set `ENQUEUE_RESOURCES` to a value higher than `DML_LOCKS + 20`, then Oracle uses the value you provide. Oracle will automatically allocate additional enqueues from the shared pool as needed if the number specified by `ENQUEUE_RESOURCES` is exceeded. You can check resource usage by querying `V$RESOURCE_LIMIT`.

See Also:

- *Oracle Real Application Clusters Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*

EVENT

Property	Description
Parameter type	String
Default value	There is no default value.
Modifiable	No
Basic	No

EVENT is a parameter used only to debug the system. Do not alter the value of this parameter except under the supervision of Oracle Support Services staff.

FAL_CLIENT

Property	Description
Parameter type	String
Syntax	<code>FAL_CLIENT = string</code>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

FAL_CLIENT specifies the FAL (fetch archive log) client name that is used by the FAL service, configured through the FAL_SERVER parameter, to refer to the FAL client. The value is an Oracle Net service name, which is assumed to be configured properly on the FAL server system to point to the FAL client (standby database).

Given the dependency of FAL_CLIENT on FAL_SERVER, the two parameters should be configured or changed at the same time.

See Also: *Oracle Data Guard Concepts and Administration* for more information about FAL server

FAL_SERVER

Property	Description
Parameter type	String
Syntax	<code>FAL_SERVER = string</code>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

FAL_SERVER specifies the FAL (fetch archive log) server for a standby database. The value is an Oracle Net service name, which is assumed to be configured properly on the standby database system to point to the desired FAL server.

See Also: *Oracle Data Guard Concepts and Administration* for more information about FAL server

FAST_START_IO_TARGET

Property	Description
Parameter type	Integer
Default value	All the buffers in the cache
Modifiable	ALTER SYSTEM
Range of values	1000 to all buffers in the cache. A setting of 0 disables limiting recovery I/Os.
Basic	No
Real Application Clusters	Multiple instances can have different values, and you can change the values at runtime.

Note: This parameter is deprecated in favor of the FAST_START_MTTTR_TARGET parameter. Oracle recommends that you use FAST_START_MTTTR_TARGET instead. FAST_START_IO_TARGET is retained for backward compatibility only.

FAST_START_IO_TARGET (available only with the Oracle Enterprise Edition) specifies the number of I/Os that should be needed during crash or instance recovery.

When you set this parameter, DBWn writes dirty buffers out more aggressively, so that the number of blocks that must be processed during recovery stays below the value specified in the parameter. However, this parameter does not impose a hard limit on the number of recovery I/Os. Under transient workload situations, the number of I/Os needed during recovery may be greater than the value specified in this parameter. In such situations, DBWn will not slow down database activity.

Smaller values for this parameter result in faster recovery times. This improvement in recovery performance is achieved at the expense of additional writing activity during normal processing.

Setting this parameter to 0 disables fast-start checkpointing, which is the mechanism that limits the number of I/Os that need to be performed during recovery. All other writing activity is unaffected.

Notes: Recovery I/O can also be limited by setting the LOG_CHECKPOINT_INTERVAL or LOG_CHECKPOINT_TIMEOUT parameter or by the size specified for the smallest redo log. For information on which mechanism is controlling checkpointing behavior, query the V\$INSTANCE_RECOVERY view.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- [V\\$INSTANCE_RECOVERY](#) on page 4-70 for information on fast-start checkpointing

FAST_START_MTTR_TARGET

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to 3600 seconds
Basic	No
Real Application Clusters	Multiple instances can have different values, and you can change the values at runtime.

FAST_START_MTTR_TARGET enables you to specify the number of seconds the database takes to perform crash recovery of a single instance. When specified, FAST_START_MTTR_TARGET

- Is overridden by FAST_START_IO_TARGET
- Is overridden by LOG_CHECKPOINT_INTERVAL

FAST_START_PARALLEL_ROLLBACK

Property	Description
Parameter type	String
Syntax	FAST_START_PARALLEL_ROLLBACK = { HI LO FALSE }
Default value	LOW
Modifiable	ALTER SYSTEM
Basic	No

FAST_START_PARALLEL_ROLLBACK determines the maximum number of processes that can exist for performing parallel rollback. This parameter is useful on systems in which some or all of the transactions are long running.

Values:

- FALSE indicates that parallel rollback is disabled
- LOW limits the number of rollback processes to 2 * CPU_COUNT
- HIGH limits the number of rollback processes to 4 * CPU_COUNT

FILE_MAPPING

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

`FILE_MAPPING` enables or disables file mapping. The FMON background process will be started to manage the mapping information when file mapping is enabled.

FILEIO_NETWORK_ADAPTERS

Property	Description
Parameter type	String
Syntax	<code>FILEIO_NETWORK_ADAPTERS = adapter_name [, adapter_name] ...</code>
Default value	There is no default value.
Modifiable	No
Range of values	One or more network adapter names, separated by commas
Basic	No

`FILEIO_NETWORK_ADAPTERS` specifies a list of network adapters that can be used to access the disk storage. On platforms where the database files reside in network attached storage, this parameter provides the storage access library the list of network adapters that can be used to access the storage.

The network adapter name is a fully qualified address name of the network card that can be accessed through the host name database or using the Network Information Service. The components of the adapter name are separated by periods. For example, the following is a fully qualified adapter name:

```
ib1.oracle.com
```

FILESYSTEMIO_OPTIONS

Property	Description
Parameter type	String
Syntax	<code>FILESYSTEMIO_OPTIONS = { none setall directIO asynch }</code>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

FILESYSTEMIO_OPTIONS specifies I/O operations for file system files.

FIXED_DATE

Property	Description
Parameter type	String
Syntax	FIXED_DATE = YYYY-MM-DD-HH24:MI:SS (or the default Oracle date format)
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

FIXED_DATE enables you to set a constant date that SYSDATE will always return instead of the current date. This parameter is useful primarily for testing. The value can be in the format shown above or in the default Oracle date format, without a time.

GC_FILES_TO_LOCKS

Property	Description
Parameter type	String
Syntax	GC_FILES_TO_LOCKS = '{file_list=lock_count[!blocks][EACH][:...]}' Spaces are not allowed within the quotation marks.
Default value	There is no default value.
Modifiable	No
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have identical values. To change the value, you must shut down all instances in the cluster, change the value for each instance, and then start up each instance.

Note: Setting this parameter to any value other than the default will disable Cache Fusion processing in a Real Application Clusters environment.

GC_FILES_TO_LOCKS is a Real Application Clusters parameter that has no effect on an instance running in exclusive mode. It controls the mapping of pre-release 9.0.1 parallel cache management (PCM) locks to datafiles.

Values:

- *file_list*

One or more datafiles listed by their file numbers, or ranges of file numbers, with comma separators:

```
filenumber[-filenumber][,filenumber[-filenumber]]...
```

To find the correspondence between filenames and file numbers, query the `FILE_NAME` and `FILE_ID` columns of the `DBA_DATA_FILES` data dictionary view.

- *lock_count*

The number of PCM locks assigned to *file_list*. By default these locks are fixed. If you set *lock_count* to 0, then Oracle uses fine-grain locking for these files and takes locks as needed from the pool of releasable locks.

- *blocks*

Specifies the number of contiguous blocks covered by one lock. The default is noncontiguous blocks.

- `EACH`

Indicates that each datafile in *file_list* is assigned a separate set of *lock_count* PCM locks.

The value of the parameter should be set to cover as many files as possible. Therefore, to avoid performance problems, you should always change `GC_FILES_TO_LOCKS` when the size of datafiles change or when new datafiles are added. Doing so requires you to shut down and restart your cluster database.

If the number of PCM locks allocated to a datafile is less than or equal to the number of blocks in a datafile, each of these locks will cover a number of contiguous blocks within the datafile equal to *blocks*. If the number of PCM locks assigned to the datafile is larger than its number of blocks, resources will be wasted because some locks will not be covering any blocks.

A colon (:) separates each clause that assigns a number of PCM locks to *file_list*.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide* and *Oracle Real Application Clusters Deployment and Performance Guide* for more information on setting this parameter

GCS_SERVER_PROCESSES

Property	Description
Parameter type	Integer
Default value	If <code>CLUSTER_DATABASE</code> is set to <code>true</code> , then 2 If <code>CLUSTER_DATABASE</code> is set to <code>false</code> , then 0
Modifiable	No
Range of values	1 to 20
Basic	No
Real Application Clusters	Multiple instances can have different values.

`GCS_SERVER_PROCESSES` specifies the initial number of server processes in Global Cache Service to serve the inter-instance traffic among Real Application Clusters instances.

GLOBAL_CONTEXT_POOL_SIZE

Property	Description
Parameter type	String
Default value	1 MB
Modifiable	No
Range of values	Any integer value in MB
Basic	No

GLOBAL_CONTEXT_POOL_SIZE specifies the amount of memory to allocate in the SGA for storing and managing global application context.

GLOBAL_NAMES

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false
Basic	No

GLOBAL_NAMES specifies whether a database link is required to have the same name as the database to which it connects.

If the value of GLOBAL_NAMES is false, then no check is performed. If you use or plan to use distributed processing, then Oracle recommends that you set this parameter to true to ensure the use of consistent naming conventions for databases and links in a networked environment.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter

HASH_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	Derived: 2 * SORT_AREA_SIZE
Modifiable	ALTER SESSION
Range of values	0 to operating system-dependent
Basic	No

Note: Oracle does not recommend using the `HASH_AREA_SIZE` parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting `PGA_AGGREGATE_TARGET` instead. `HASH_AREA_SIZE` is retained for backward compatibility.

`HASH_AREA_SIZE` is relevant to parallel execution operations and to the query portion of DML or DDL statements. It specifies the maximum amount of memory, in bytes, to be used for hash joins.

See Also:

- *Oracle Database Concepts* for information on hash joins in general
- *Oracle Database Performance Tuning Guide* for information on calculating an appropriate value for this parameter

HI_SHARED_MEMORY_ADDRESS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Basic	No

`HI_SHARED_MEMORY_ADDRESS` specifies the starting address at runtime of the system global area (SGA). It is ignored on platforms that specify the SGA's starting address at linktime.

On 64-bit platforms, use `HI_SHARED_MEMORY_ADDRESS` to specify the high-order 32 bits of a 64-bit address. Use `SHARED_MEMORY_ADDRESS` to specify the low-order 32 bits of the address (see "[SHARED_MEMORY_ADDRESS](#)" on page 1-127). If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

HS_AUTOREGISTER

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

`HS_AUTOREGISTER` enables or disables automatic self-registration of Heterogeneous Services (HS) agents. When enabled, information is uploaded into the server's data dictionary to describe a previously unknown agent class or a new agent version.

Oracle recommends that you set this parameter to `true`. Oracle incurs less overhead when establishing subsequent connections through the same agent if self-registered information is available in the server’s data dictionary.

See Also: *Oracle Database Heterogeneous Connectivity Administrator’s Guide* for more information on HS agents

IFILE

Property	Description
Parameter type	Parameter file
Syntax	<code>IFILE = parameter_file_name</code>
Default value	There is no default value.
Modifiable	No
Range of values	Valid parameter filenames
Basic	No
Real Application Clusters	Multiple instances can have different values.

Use `IFILE` to embed another parameter file within the current parameter file. For example:

```
IFILE = COMMON.ORA
```

You can have up to three levels of nesting. In this example, the file `COMMON.ORA` could contain a second `IFILE` parameter for the file `COMMON2.ORA`, which could contain a third `IFILE` parameter for the file `GCPARMS.ORA`. You can also include multiple parameter files in one parameter file by listing `IFILE` several times with different values:

```
IFILE = DBPARMS.ORA
IFILE = GCPARMS.ORA
IFILE = LOGPARMS.ORA
```

Note: You must list multiple entries on contiguous lines of the parameter file.

INSTANCE_GROUPS

Property	Description
Parameter type	String
Syntax	<code>INSTANCE_GROUPS = group_name [, group_name] ...</code>
Default value	There is no default value.
Modifiable	No
Range of values	One or more instance group names, separated by commas
Basic	No

Property	Description
Real Application Clusters	Multiple instances can have different values.

INSTANCE_GROUPS is a Real Application Clusters parameter that you can specify only in parallel mode. Used in conjunction with the PARALLEL_INSTANCE_GROUP parameter, it lets you restrict parallel query operations to a limited number of instances.

This parameter specifies one or more instance groups and assigns the current instance to those groups. If one of the specified groups is also specified in the PARALLEL_INSTANCE_GROUP parameter, then Oracle allocates query processes for a parallel operation from this instance.

See Also:

- *Oracle Real Application Clusters Administrator's Guide* for more information on parallel query execution in a Real Application Clusters environment
- "[PARALLEL_INSTANCE_GROUP](#)" on page 1-103

INSTANCE_NAME

Property	Description
Parameter type	String
Syntax	INSTANCE_NAME = <i>instance_id</i>
Default value	The instance's SID Note: The SID identifies the instance's shared memory on a host, but may not uniquely distinguish this instance from other instances.
Modifiable	No
Range of values	Any alphanumeric characters
Basic	No

In a Real Application Clusters environment, multiple instances can be associated with a single database service. Clients can override Oracle's connection load balancing by specifying a particular instance by which to connect to the database. INSTANCE_NAME specifies the unique name of this instance.

In a single-instance database system, the instance name is usually the same as the database name.

See Also: *Oracle Real Application Clusters Administrator's Guide* and *Oracle Net Services Administrator's Guide* for more information

INSTANCE_NUMBER

Property	Description
Parameter type	Integer

Property	Description
Default value	Lowest available number; derived from instance start up order and INSTANCE_NUMBER value of other instances. If not configured for Real Application Clusters, then 0.
Modifiable	No
Range of values	1 to maximum number of instances specified when the database was created
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and all instances must have different values.

INSTANCE_NUMBER is a Real Application Clusters parameter that can be specified in parallel mode or exclusive mode. It specifies a unique number that maps the instance to one free list group for each database object created with storage parameter FREELIST GROUPS.

The INSTANCE parameter of the ALTER TABLE ... ALLOCATE EXTENT statement assigns an extent to a particular free list group. If you set INSTANCE_NUMBER to the value specified for the INSTANCE parameter, the instance uses that extent for inserts and for updates that expand rows.

The practical maximum value of this parameter is the maximum number of instances specified in the CREATE DATABASE statement. The absolute maximum is operating system-dependent.

See Also: *Oracle Real Application Clusters Administrator's Guide* for more information

INSTANCE_TYPE

Property	Description
Parameter type	String
Syntax	INSTANCE_TYPE = { RDBMS ASM }
Default value	RDBMS
Modifiable	No
Basic	No
Real Application Clusters	Multiple instances must have the same value.

INSTANCE_TYPE specifies whether the instance is a database instance or an Automatic Storage Management instance.

Values:

- RDBMS
The instance is a database instance.
- ASM
The instance is an Automatic Storage Management instance.

JAVA_MAX_SESSIONSPACE_SIZE

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. `JAVA_MAX_SESSIONSPACE_SIZE` specifies (in bytes) the maximum amount of session space made available to a Java program executing in the server. When a user's session-duration Java state attempts to exceed this amount, the Java virtual machine kills the session with an out-of-memory failure.

See Also:

- *Oracle Database Java Developer's Guide*
- "[JAVA_SOFT_SESSIONSPACE_LIMIT](#)" on page 1-64

JAVA_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>JAVA_POOL_SIZE = integer [K M G]</code>
Default value	If <code>SGA_TARGET</code> is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If <code>SGA_TARGET</code> is not set: 24 MB, rounded up to the nearest granule size
Modifiable	<code>ALTER SYSTEM</code>
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

`JAVA_POOL_SIZE` specifies (in bytes) the size of the Java pool, from which the Java memory manager allocates most Java state during runtime execution. This memory includes the shared in-memory representation of Java method and class definitions, as well as the Java objects that are migrated to the Java session space at end-of-call.

See Also: *Oracle Database Java Developer's Guide* for information on adjusting this parameter

JAVA_SOFT_SESSIONSPACE_LIMIT

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. `JAVA_SOFT_SESSIONSPACE_LIMIT` specifies (in bytes) a **soft limit** on Java memory usage in a session, as a means to warn you if a user's session-duration Java state is using too much memory. When a user's session-duration Java state exceeds this size, Oracle generates a warning that goes into the trace files.

See Also:

- *Oracle Database Java Developer's Guide* for more information on this parameter
- ["JAVA_MAX_SESSIONSPACE_SIZE"](#) on page 1-63

JOB_QUEUE_PROCESSES

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to 1000
Basic	Yes
Real Application Clusters	Multiple instances can have different values.

`JOB_QUEUE_PROCESSES` specifies the maximum number of processes that can be created for the execution of jobs. It specifies the number of job queue processes per instance (J000, ... J999). Replication uses job queues for data refreshes. Advanced queuing uses job queues for message propagation. You can create user job requests through the `DBMS_JOB` package.

Some job queue requests are created automatically. An example is refresh support for materialized views. If you wish to have your materialized views updated automatically, you must set `JOB_QUEUE_PROCESSES` to a value of one or higher.

See Also:

- *Oracle Database Advanced Replication and Oracle Data Warehousing Guide* for more information on managing materialized views
- *Oracle Streams Advanced Queuing User's Guide and Reference* for more information on message propagation

LARGE_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>LARGE_POOL_SIZE = integer [K M G]</code>
Default value	0 if both of the following are true: <ul style="list-style-type: none"> ■ The pool is not required by parallel execution ■ <code>DBWR_IO_SLAVES</code> is not set Otherwise, derived from the values of <code>PARALLEL_MAX_SERVERS</code> , <code>PARALLEL_THREADS_PER_CPU</code> , <code>CLUSTER_DATABASE_INSTANCES</code> , <code>DISPATCHERS</code> , and <code>DBWR_IO_SLAVES</code> .
Modifiable	<code>ALTER SYSTEM</code>
Range of values	300 KB to at least 2 GB (actual maximum is operating system-specific)
Basic	No

`LARGE_POOL_SIZE` specifies (in bytes) the size of the large pool allocation heap. The large pool allocation heap is used in shared server systems for session memory, by parallel execution for message buffers, and by backup processes for disk I/O buffers. Parallel execution allocates buffers out of the large pool only when `PARALLEL_AUTOMATIC_TUNING` is set to `true`.

Caution: When Oracle derives a default value, it adds 250K for each session for the shared server if `DISPATCHERS` is configured. The final derived value also includes a port-specific amount of memory for backup I/O buffers. The total derived default value can either be too large to allocate or can cause performance problems. In that case, set `LARGE_POOL_SIZE` to a number sufficiently small so that the database can start.

You can specify the value of this parameter using a number, optionally followed by K or M to specify kilobytes or megabytes, respectively. If you do not specify K or M, then the number is taken as bytes.

See Also: *Oracle Database Performance Tuning Guide* and *Oracle Database Upgrade Guide* for more information on setting this parameter

LDAP_DIRECTORY_ACCESS

Property	Description
Parameter type	String
Syntax	<code>LDAP_DIRECTORY_ACCESS = { NONE PASSWORD SSL }</code>
Default value	NONE
Modifiable	<code>ALTER SYSTEM</code>
Basic	No

LDAP_DIRECTORY_ACCESS specifies whether Oracle refers to Oracle Internet Directory for user authentication information. If directory access is turned on, then this parameter also specifies how users are authenticated.

Values:

- NONE
Oracle does not refer to Oracle Internet Directory for Enterprise User Security information.
- PASSWORD
Oracle tries to connect to the enterprise directory service using the database password stored in the database wallet. If that fails, then the Oracle Internet Directory connection fails and the database will not be able to retrieve enterprise roles and schema mappings upon enterprise user login.
- SSL
Oracle tries to connect to Oracle Internet Directory using SSL.

See Also: *Oracle Advanced Security Administrator's Guide* for more information on Enterprise User Security

LICENSE_MAX_SESSIONS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to number of session licenses
Basic	No
Real Application Clusters	Multiple instances can have different values, but the total for all instances mounting a database should be less than or equal to the total number of sessions licensed for that database.

LICENSE_MAX_SESSIONS specifies the maximum number of concurrent user sessions allowed. When this limit is reached, only users with the RESTRICTED SESSION privilege can connect to the database. Users who are not able to connect receive a warning message indicating that the system has reached maximum capacity.

A zero value indicates that concurrent usage (session) licensing is not enforced. If you set this parameter to a nonzero number, you might also want to set LICENSE_SESSIONS_WARNING (see "[LICENSE_SESSIONS_WARNING](#)" on page 1-67).

Do not enable both concurrent usage licensing and user licensing. Set either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS to zero.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter

LICENSE_MAX_USERS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to number of user licenses
Basic	No
Real Application Clusters	Multiple instances should have the same values. If different instances specify different values for this parameter, then the value of the first instance to mount the database takes precedence.

`LICENSE_MAX_USERS` specifies the maximum number of users you can create in the database. When you reach this limit, you cannot create more users. You can, however, increase the limit.

Do not enable both concurrent usage (session) licensing and user licensing. Set either `LICENSE_MAX_SESSIONS` or `LICENSE_MAX_USERS` to zero.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter

LICENSE_SESSIONS_WARNING

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to value of <code>LICENSE_MAX_SESSIONS</code> parameter
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LICENSE_SESSIONS_WARNING` specifies a warning limit on the number of concurrent user sessions. When this limit is reached, additional users can connect, but Oracle writes a message in the alert file for each new connection. Users with `RESTRICTED SESSION` privilege who connect after the limit is reached receive a warning message stating that the system is nearing its maximum capacity.

If this parameter is set to zero, no warning is given as you approach the concurrent usage (session) limit. If you set this parameter to a nonzero number, you should also set `LICENSE_MAX_SESSIONS` (see "[LICENSE_MAX_SESSIONS](#)" on page 1-66).

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Real Application Clusters Administrator's Guide* for more information on calculating an appropriate value for this parameter

LOCAL_LISTENER

Property	Description
Parameter type	String
Syntax	LOCAL_LISTENER = <i>network_name</i>
Default value	(ADDRESS = (PROTOCOL=TCP)(HOST= <i>hostname</i>)(PORT=1521)) where <i>hostname</i> is the network name of the local host.
Modifiable	ALTER SYSTEM
Basic	No

LOCAL_LISTENER specifies a network name that resolves to an address or address list of Oracle Net local listeners (that is, listeners that are running on the same machine as this instance). The address or address list is specified in the TNSNAMES.ORA file or other address repository as configured for your system.

See Also:

- *Oracle Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle Net Services Administrator's Guide* and your operating system-specific Oracle documentation for more information about specifying network addresses for the protocols on your system

LOCK_NAME_SPACE

Property	Description
Parameter type	String
Syntax	LOCK_NAME_SPACE = <i>namespace</i>
Default value	There is no default value.
Modifiable	No
Range of values	Up to 8 alphanumeric characters. No special characters allowed.
Basic	No

LOCK_NAME_SPACE specifies the namespace that the distributed lock manager (DLM) uses to generate lock names. Consider setting this parameter if a standby or clone database has the same database name on the same cluster as the primary database.

If the standby database resides on the same file system as the primary database, then set LOCK_NAME_SPACE in the standby parameter file to a distinct value such as the following:

LOCK_NAME_SPACE = standby

LOCK_SGA

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false
Basic	No

LOCK_SGA locks the entire SGA into physical memory. It is usually advisable to lock the SGA into real (physical) memory, especially if the use of virtual memory would include storing some of the SGA using disk space. This parameter is ignored on platforms that do not support it.

LOG_ARCHIVE_CONFIG

Property	Description
Parameter type	String
Syntax	<pre>LOG_ARCHIVE_CONFIG = { [SEND NOSEND] [RECEIVE NORECEIVE] [DG_CONFIG=(remote_db_unique_name1 [, ... remote_db_ unique_name9) NODG_CONFIG] }</pre>
Default value	'SEND, RECEIVE, NODG_CONFIG'
Modifiable	ALTER SYSTEM
Basic	No

LOG_ARCHIVE_CONFIG enables or disables the sending of redo logs to remote destinations and the receipt of remote redo logs, and specifies the unique database names (DB_UNIQUE_NAME) for each database in the Data Guard configuration.

Values:

- SEND
Enables the sending of redo logs to remote destinations
- NOSEND
Disables the sending of redo logs to remote destinations
- RECEIVE
Enables the receipt of remotely archived redo logs
- NORECEIVE

Disables the receipt of remotely archived redo logs

- DG_CONFIG

Specifies a list of up to 9 unique database names (defined with the DB_UNIQUE_NAME initialization parameter) for all of the databases in the Data Guard configuration.

- NODG_CONFIG

Eliminates the list of service provider names previously specified with the DG_CONFIG option.

LOG_ARCHIVE_DEST

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DEST = <i>filespec</i>
Default value	Null
Modifiable	ALTER SYSTEM
Range of values	Any valid path or device name, except raw partitions
Basic	No
Real Application Clusters	Multiple instances can have different values.

Note: For Enterprise Edition users, this parameter has been deprecated in favor of the LOG_ARCHIVE_DEST_1 parameters. If Oracle Enterprise Edition is not installed or it is installed, but you have not specified any LOG_ARCHIVE_DEST_1 parameters, this parameter is valid.

LOG_ARCHIVE_DEST is applicable only if you are running the database in ARCHIVELOG mode or are recovering a database from archived redo logs. LOG_ARCHIVE_DEST is incompatible with the LOG_ARCHIVE_DEST_1 parameters, and must be defined as the null string ("" or ' ') when any LOG_ARCHIVE_DEST_1 parameter has a value other than a null string. Use a text string to specify the default location and root of the disk file or tape device when archiving redo log files. (Archiving to tape is not supported on all operating systems.) The value cannot be a raw partition.

If LOG_ARCHIVE_DEST is not explicitly defined and all the LOG_ARCHIVE_DEST_1 parameters have null string values, LOG_ARCHIVE_DEST is set to an operating system-specific default value on instance startup.

To override the destination that this parameter specifies, either specify a different destination for manual archiving or use the SQL*Plus statement ARCHIVE LOG START *filespec* for automatic archiving, where *filespec* is the new archive destination. To permanently change the destination, use the statement ALTER SYSTEM SET LOG_ARCHIVE_DEST = *filespec*, where *filespec* is the new archive destination.

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the

variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

See Also:

- *Oracle Database Backup and Recovery Advanced User's Guide*
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-73, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-76, and "V\$ARCHIVE_DEST" on page 4-6 for more information on setting this parameter
- Your Oracle operating system-specific documentation for the default value and for an example of how to specify the destination path or filename using LOG_ARCHIVE_DEST

LOG_ARCHIVE_DEST_n

Property	Description
Parameter type	String
Syntax	<pre>LOG_ARCHIVE_DEST_{1 2 3 4 5 6 7 8 9 10} = { null_string { LOCATION=path_name SERVICE=service_name } [{ MANDATORY OPTIONAL }] [REOPEN[=seconds] NOREOPEN] [DELAY[=minutes] NODELAY] [REGISTER[=template] NOREGISTER] [TEMPLATE=template] NOTEMPLATE] [ALTERNATE=destination NOALTERNATE] [DEPENDENCY=destination NODEPENDENCY] [MAX_FAILURE=count NOMAX_FAILURE] [QUOTA_SIZE=blocks NOQUOTA_SIZE] [QUOTA_USED=blocks NOQUOTA_USED] [ARCH LGWR] [SYNC[=PARALLEL NOPARALLEL] ASYNC[=blocks]] [AFFIRM NOAFFIRM] [NET_TIMEOUT=seconds NONET_TIMEOUT] [VALID_FOR=(redo_log_type,database_role)] [DB_UNIQUE_NAME NODB_UNIQUE_NAME] [VERIFY NOVERIFY] }</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	Yes

Note: This parameter is valid only if you have installed the Oracle Enterprise Edition. You may continue to use LOG_ARCHIVE_DEST if you have installed the Oracle Enterprise Edition. However, you cannot use both LOG_ARCHIVE_DEST_n and LOG_ARCHIVE_DEST, as they are not compatible.

The LOG_ARCHIVE_DEST_1 parameters (where $n = 1, 2, 3, \dots, 10$) define up to ten archive log destinations. The parameter integer suffix is defined as the **handle** displayed by the V\$ARCHIVE_DEST dynamic performance view.

Values:

- SERVICE
Specifies a standby destination. Oracle Net (IPC or TCP) transmits the archive log. A standby instance must be associated with the destination. The value represented by *tnsnames_service* corresponds to an appropriate service name in *tnsnames.ora*.
- LOCATION
Specifies a local file system destination. You must specify this parameter for at least one destination.
- MANDATORY
Specifies that archiving to the destination must succeed before the redo log file can be made available for reuse.
- OPTIONAL
Specifies that successful archiving to the destination is not required before the redo log file can be made available for reuse. If the "must succeed count," set with LOG_ARCHIVE_MIN_SUCCEED_DEST, is met, the redo logfile is marked for reuse. This is the default.
- REOPEN
Specifies the minimum number of seconds before the archiver process (*ARC_n*, foreground, or log writer process) should try again to access a previously failed destination. Future attempts are made when the next redo log file is archived. If a destination is MANDATORY, then Oracle recommends that you specify a REOPEN time that reduces the possibility of primary database shutdown due to lack of available online redo log files.

If you do not specify *seconds*, then the default value is 300 seconds.

It is recommended that the destination be explicitly enabled once it is dynamically set. For example, after defining a LOG_ARCHIVE_DEST_1 parameter, an ALTER [SESSION | SYSTEM] SET LOG_ARCHIVE_DEST_STATE_1 statement should be issued to properly enable the destination.

See Also: *Oracle Data Guard Concepts and Administration* and *Oracle Database Administrator's Guide* for more information and examples

LOG_ARCHIVE_DEST_STATE_n

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DEST_STATE_n = { alternate reset defer enable }
Default value	enable
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	Yes

The LOG_ARCHIVE_DEST_STATE_ *n* parameters (where *n* = 1, 2, 3, ... 10) specify the availability state of the corresponding destination. The parameter suffix (1 through 10) specifies one of the ten corresponding LOG_ARCHIVE_DEST_ *n* destination parameters.

Values:

- `enabled`
Specifies that a valid log archive destination can be used for a subsequent archiving operation (automatic or manual). This is the default.
- `defer`
Specifies that valid destination information and attributes are preserved, but the destination is excluded from archiving operations until re-enabled.
- `alternate`
Specifies that a log archive destination is not enabled but will become enabled if communications to another destination fail.

The LOG_ARCHIVE_DEST_STATE_ *n* parameters have no effect on the ENABLE state for the LOG_ARCHIVE_DEST or LOG_ARCHIVE_DUPLEX_DEST parameters.

The V\$ARCHIVE_DEST dynamic performance view shows values in use for the current session. The DEST_ID column of that view corresponds to the archive destination suffix *n*.

See Also:

- *Oracle Database Backup and Recovery Advanced User's Guide* and *Oracle Database Administrator's Guide* for more information and examples
- "`V$ARCHIVE_DEST`" on page 4-6

LOG_ARCHIVE_DUPLEX_DEST

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DUPLEX_DEST = <i>filespec</i>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	Either a null string or any valid path or device name, except raw partitions
Basic	No

Note: If you are using Oracle Enterprise Edition, this parameter is deprecated in favor of the LOG_ARCHIVE_DEST_ *n* parameters. If Oracle Enterprise Edition is not installed or it is installed but you have not specified any LOG_ARCHIVE_DEST_ *n* parameters, this parameter is valid.

LOG_ARCHIVE_DUPLEX_DEST is similar to the initialization parameter LOG_ARCHIVE_DEST. This parameter specifies a second archive destination: the **duplex** archive destination. This duplex archive destination can be either a must-succeed or a best-effort archive destination, depending on how many archive destinations must succeed (as specified in the LOG_ARCHIVE_MIN_SUCCEED_DEST parameter).

The default setting of a null string ("") or (' ') indicates that a duplex archive destination does not exist.

See Also:

- "LOG_ARCHIVE_DEST_n" on page 1-71
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-76
- "V\$ARCHIVE_DEST" on page 4-6

LOG_ARCHIVE_FORMAT

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_FORMAT = <i>filename</i>
Default value	Operating system-dependent
Modifiable	No
Range of values	Any string that resolves to a valid filename
Basic	No
Real Application Clusters	Multiple instances can have different values, but identical values are recommended.

LOG_ARCHIVE_FORMAT is applicable only if you are using the redo log in ARCHIVELOG mode. Use a text string and variables to specify the default filename format when archiving redo log files. The string generated from this format is appended to the string specified in the LOG_ARCHIVE_DEST parameter.

The following variables can be used in the format:

%s log sequence number

%S log sequence number, zero filled

%t thread number

%T thread number, zero filled

%a activation ID

%d database ID

%r resetlogs ID that ensures unique names are constructed for the archived log files across multiple incarnations of the database

Using uppercase letters for the variables (for example, %S) causes the value to be fixed length and padded to the left with zeros. An example of specifying the archive redo log filename format follows:

```
LOG_ARCHIVE_FORMAT = 'log%t_%s_%r.arc'
```

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

See Also:

- *Oracle Database Backup and Recovery Basics, Oracle Data Guard Concepts and Administration, and Oracle Real Application Clusters Administrator's Guide* for more information on setting this parameter
- Your operating system- specific Oracle documentation for the default value and range of values for LOG_ARCHIVE_FORMAT

LOG_ARCHIVE_LOCAL_FIRST

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

LOG_ARCHIVE_LOCAL_FIRST specifies when the archiver processes (ARC n) transmit redo data to remote standby database destinations.

Values:

- true

Directs the ARC n process to transmit redo data after the online redo log file has been completely and successfully archived to at least one local destination. This is the default value.

Because the online redo log files are archived locally first, the LGWR process reuses the online redo log files much earlier than would be possible if the ARC n processes archived to the standby database concurrently with the local destination. This behavior is useful when archiving to remote destinations that use a slow network connection, such as a long-distance wide area network (WAN).

- false

Directs the ARC n process to transmit redo data at the same time the online redo log file is archived to the local destinations. This results in redo data being promptly dispatched to the remote standby database destination.

Setting LOG_ARCHIVE_LOCAL_FIRST to false is most useful for faster network connections, such as high-speed local area networks (LAN).

If LOG_ARCHIVE_LOCAL_FIRST is set to true, then it is not used on a physical standby database and any database for which the following attributes have been specified in the LOG_ARCHIVE_DEST_ n initialization parameter:

- MANDATORY
- LOCAL

- **DEPENDENCY** (or any database destination that is the target of a **DEPENDENCY** attribute)

If `LOG_ARCHIVE_LOCAL_FIRST` is set to `true`, then it is ignored during certain operations, such as during a switchover, which requires synchronized archival operations. If the destination was explicitly configured to use the log writer process (by specifying the `LGWR` attribute in the `LOG_ARCHIVE_DEST_n` initialization parameter), but for some reason the log writer process becomes unable to archive to the destination, then Data Guard will revert to using the `ARCn` process to complete archival operations using the default behavior, even if `LOG_ARCHIVE_LOCAL_FIRST` is set to `false`.

For example, if a standby database problem or a network problem causes the `LGWR` process to fail, then the `ARCn` process will complete the archival. Data Guard minimizes the effect on the primary database as much as possible by archiving to the local destination first to ensure the online redo log files are available to the `LGWR` process as quickly as possible.

See Also: *Oracle Data Guard Concepts and Administration* for more information about the `LOG_ARCHIVE_LOCAL_FIRST` initialization parameter and its effect on `ARCn` behavior

LOG_ARCHIVE_MAX_PROCESSES

Property	Description
Parameter type	Integer
Default value	2
Modifiable	ALTER SYSTEM
Range of values	1 to 10
Basic	No

`LOG_ARCHIVE_MAX_PROCESSES` specifies the number of archiver background processes (`ARC0` through `ARC9`) Oracle initially invokes.

- If the `LOG_ARCHIVE_START` initialization parameter has the value `true`, then this value is evaluated at instance startup.
- Otherwise, this parameter is evaluated when the archiver process is first invoked by `SQL*Plus` or `SQL` syntax.

The actual number of archiver processes in use may vary subsequently based on archive workload.

See Also: *Oracle Database Concepts*

LOG_ARCHIVE_MIN_SUCCEED_DEST

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM

Property	Description
Range of values	1 to 10 if you are using LOG_ARCHIVE_DEST_n 1 or 2 if you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST
Basic	No

LOG_ARCHIVE_MIN_SUCCEED_DEST defines the minimum number of destinations that must succeed in order for the online logfile to be available for reuse.

- If you are using the LOG_ARCHIVE_DEST_n parameters and automatic archiving is enabled, the value of this parameter cannot exceed either of the following:
 - The total number of destinations
 - The number of enabled, valid destinations specified as MANDATORY plus the number of enabled, valid non-standby destinations specified as OPTIONAL
- If you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST and automatic archiving is enabled, a value of 1 specifies that the destination specified in LOG_ARCHIVE_DEST must succeed. A value of 2 specifies that the destinations specified in both parameters must succeed.

If the value of this parameter is less than the number of enabled, valid MANDATORY destinations, this parameter is ignored in favor of the MANDATORY destination count. If the value is more than the number of enabled, valid MANDATORY destinations, some of the enabled, valid OPTIONAL non-standby destinations are treated as MANDATORY.

You cannot set a value for this parameter dynamically if LOG_ARCHIVE_DEST or LOG_ARCHIVE_DUPLEX_DEST are in use. However, you can switch dynamically from one system to the other using ALTER SYSTEM, as follows:

1. Set LOG_ARCHIVE_MIN_SUCCEED_DEST to 1.
2. Set the value of LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST to the null string.
3. Set the desired number of destinations for the LOG_ARCHIVE_DEST_n parameters.
4. Reset LOG_ARCHIVE_MIN_SUCCEED_DEST to the desired value.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- "[LOG_ARCHIVE_DEST_n](#)" on page 1-71, "[LOG_ARCHIVE_DUPLEX_DEST](#)" on page 1-73, and "[V\\$ARCHIVE_DEST](#)" on page 4-6 for information on related parameters

LOG_ARCHIVE_TRACE

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM

Property	Description
Range of values	0, 1, 2, 4, 8, 16, 32, 64, 128
Basic	No
Real Application Clusters	Multiple instances can have different values.

LOG_ARCHIVE_TRACE controls output generated by the archivelog process.

This process can be initiated by any of the following:

- An ARC n background process (designated as ARC n in the output logs)
- An explicit session-invoked foreground process (designated as ARCH in the output logs)

The valid values have the following meanings:

- 0: Disable archivelog tracing (this is the default)
- 1: Track archival of redo log file
- 2: Track archival status of each archivelog destination
- 4: Track archival operational phase
- 8: Track archivelog destination activity
- 16: Track detailed archivelog destination activity
- 32: Track archivelog destination parameter modifications
- 64: Track ARC n process state activity
- 128: Track FAL (fetch archived log) server related activities

You can combine tracing levels by adding together the values of the desired tracing levels. For example, a setting of 3 will generate level 1 and level 2 trace output. You can set different values for the primary and standby database.

When this parameter is set to the default value of 0, Oracle will still generate appropriate alert and trace entries in response to error conditions. If you change the value of this parameter dynamically in an ALTER SYSTEM statement, the changes will take effect at the start of the next archivelog operation.

See Also: *Oracle Database Administrator's Guide*

LOG_BUFFER

Property	Description
Parameter type	Integer
Default value	512 KB or 128 KB * CPU_COUNT, whichever is greater
Modifiable	No
Range of values	Operating system-dependent
Basic	No

LOG_BUFFER specifies the amount of memory (in bytes) that Oracle uses when buffering redo entries to a redo log file. Redo log entries contain a record of the

changes that have been made to the database block buffers. The LGWR process writes redo log entries from the log buffer to a redo log file.

In general, larger values for LOG_BUFFER reduce redo log file I/O, particularly if transactions are long or numerous. In a busy system, a value 65536 or higher is reasonable.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

LOG_CHECKPOINT_INTERVAL

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	Unlimited
Basic	No
Real Application Clusters	Multiple instances can have different values.

LOG_CHECKPOINT_INTERVAL specifies the frequency of checkpoints in terms of the number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log. This number refers to physical operating system blocks, not database blocks.

Regardless of this value, a checkpoint always occurs when switching from one online redo log file to another. Therefore, if the value exceeds the actual redo log file size, checkpoints occur only when switching logs. Checkpoint frequency is one of the factors that influence the time required for the database to recover from an unexpected failure.

Notes:

- Specifying a value of 0 (zero) for LOG_CHECKPOINT_INTERVAL has the same effect as setting the parameter to infinity and causes the parameter to be ignored. Only nonzero values of this parameter are considered meaningful.
- Recovery I/O can also be limited by setting the FAST_START_IO_TARGET or LOG_CHECKPOINT_TIMEOUT parameter or by the size specified for the smallest redo log. For information on which mechanism is controlling checkpointing behavior, query the V\$INSTANCE_RECOVERY view.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "[FAST_START_IO_TARGET](#)" on page 1-53, "[LOG_CHECKPOINT_TIMEOUT](#)" on page 1-80, and "[V\\$INSTANCE_RECOVERY](#)" on page 4-70

LOG_CHECKPOINT_TIMEOUT

Property	Description
Parameter type	Integer
Default value	1800
Modifiable	ALTER SYSTEM
Range of values	0 to unlimited
Basic	No
Real Application Clusters	Multiple instances can have different values.

LOG_CHECKPOINT_TIMEOUT specifies (in seconds) the amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log (sometimes called the **tail of the log**) occurred. This parameter also signifies that no buffer will remain dirty (in the cache) for more than *integer* seconds.

Specifying a value of 0 for the timeout disables time-based checkpoints. Hence, setting the value to 0 is not recommended unless FAST_START_MTTR_TARGET is set.

Notes:

- A checkpoint scheduled to occur because of this parameter is delayed until the completion of the previous checkpoint if the previous checkpoint has not yet completed.
- Recovery I/O can also be limited by setting the FAST_START_IO_TARGET or LOG_CHECKPOINT_INTERVAL parameter or by the size specified for the smallest redo log. For information on which mechanism is controlling checkpointing behavior, query the V\$INSTANCE_RECOVERY view.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "[FAST_START_IO_TARGET](#)" on page 1-53, "[LOG_CHECKPOINT_TIMEOUT](#)" on page 1-80, and "[V\\$INSTANCE_RECOVERY](#)" on page 4-70

LOG_CHECKPOINTS_TO_ALERT

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Range of values	true false
Basic	No

LOG_CHECKPOINTS_TO_ALERT lets you log your checkpoints to the alert file. Doing so is useful for determining whether checkpoints are occurring at the desired frequency.

LOG_FILE_NAME_CONVERT

Property	Description
Parameter type	String
Syntax	<pre>DB_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' , ...</pre> <p>Where:</p> <ul style="list-style-type: none"> ▪ <i>string1</i> is the pattern of the primary database filename ▪ <i>string2</i> is the pattern of the standby database filename ▪ <i>string3</i> is the pattern of the primary database filename ▪ <i>string4</i> is the pattern of the standby database filename <p>You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks.</p> <p>The following are example settings that are acceptable:</p> <pre>DB_FILE_NAME_CONVERT=' /dbs/t1/' , '/dbs/t1/s_' , 'dbs/t2/' , 'dbs/t2/s_'</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION
Basic	No

LOG_FILE_NAME_CONVERT converts the filename of a new log file on the primary database to the filename of a log file on the standby database. If you add a log file to the primary database, you must add a corresponding file to the standby database.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is signalled during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalued parameters).

When the standby database is updated, this parameter converts the log file name on the primary database to the log file name on the standby database. The file must exist on the standby database and must be writable or the recovery process will halt with an error.

The first string is the pattern found in the log file names on the primary database. The second string is the pattern found in the log file names on the standby database.

You should also use LOG_FILE_NAME_CONVERT to rename the logfiles in the clone controlfile when setting up the clone database during tablespace point-in-time recovery.

See Also: *Oracle Database Backup and Recovery Advanced User's Guide* and *Oracle Data Guard Concepts and Administration*

LOGMNR_MAX_PERSISTENT_SESSIONS

Property	Description
Parameter type	Integer
Default value	1
Modifiable	No
Range of values	1 to LICENSE_MAX_SESSIONS
Basic	No

LOGMNR_MAX_PERSISTENT_SESSIONS enables you to specify the maximum number of persistent LogMiner mining sessions (which are LogMiner sessions that are backed up on disk) that are concurrently active when all sessions are mining redo logs generated by standalone instances. This pre-allocates $2 * \text{LOGMNR_MAX_PERSISTENT_SESSIONS}$ MB of contiguous memory in the SGA for use by LogMiner.

LogMiner requires 2 MB of contiguous memory for mining the redo log stream generated by a non-cluster database instance. To mine redo logs generated by an n -instance cluster database system, 2 MB of contiguous memory is required for each redo thread (or, a total of $2 * n$). Because persistent LogMiner sessions stage all data in the SGA, read buffers must be allocated in the SGA. The required contiguous memory is allocated at instance startup.

LOGMNR_MAX_PERSISTENT_SESSIONS is normalized for redo logs generated by non-cluster database instances. If you wish to mine a redo log stream generated by a 3-node cluster database instance, you should set this parameter to 3 (because one persistent session mining redo log generated in a 3-node cluster database instance is equivalent to 3 persistent sessions, each with mining redo logs generated by a non-cluster database instance). The memory remains exclusively allocated for LogMiner persistent sessions. This parameter does not affect transient LogMiner sessions that are not backed up on disk and are staged entirely in the Process Global Area (PGA).

See Also: *PL/SQL Packages and Types Reference* for information on the DBMS_LOGMNR_SESSION package

MAX_COMMIT_PROPAGATION_DELAY

Property	Description
Parameter type	Integer
Default value	700
Modifiable	No

Property	Description
Range of values	0 to 90000
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have identical values.

Caution: Change this parameter only when it is absolutely necessary to see the most current version of the database when performing a query.

MAX_COMMIT_PROPAGATION_DELAY is a Real Application Clusters parameter. This initialization parameter should not be changed except under a limited set of circumstances specific to the cluster database.

This parameter specifies the maximum amount of time allowed before the system change number (SCN) held in the SGA of an instance is refreshed by the log writer process (LGWR). It determines whether the local SCN should be refreshed from the lock value when getting the snapshot SCN for a query. Units are in hundredths of seconds.

Under very unusual circumstances involving rapid updates and queries of the same data from different instances, the SCN might not be refreshed in a timely manner. Setting the parameter to zero causes the SCN to be refreshed immediately after a commit. The default value (700 hundredths of a second, or seven seconds) is an upper bound that allows the preferred existing high performance mechanism to remain in place.

See Also: *Oracle Real Application Clusters Administrator's Guide* for more information on setting this parameter

MAX_DISPATCHERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If MAX_DISPATCHERS is specified, then it should be greater than or equal to the number of dispatchers specified by the DISPATCHERS parameter and less than the number of processes specified by the PROCESSES parameter.
Basic	No

MAX_DISPATCHERS specifies the maximum number of dispatcher processes allowed to be running simultaneously. It can be overridden by the DISPATCHERS parameter and is maintained for backward compatibility with older releases.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

MAX_DUMP_FILE_SIZE

Property	Description
Parameter type	String
Syntax	MAX_DUMP_FILE_SIZE = { <i>integer</i> [K M] UNLIMITED }
Default value	UNLIMITED
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to unlimited, or UNLIMITED
Basic	No

MAX_DUMP_FILE_SIZE specifies the maximum size of trace files (excluding the alert file). Change this limit if you are concerned that trace files may use too much space.

- A numerical value for MAX_DUMP_FILE_SIZE specifies the maximum size in operating system blocks.
- A number followed by a K or M suffix specifies the file size in kilobytes or megabytes.
- The special value string UNLIMITED means that there is no upper limit on trace file size. Thus, dump files can be as large as the operating system permits.

See Also: *Oracle Database Administrator's Guide* and *Oracle Database Performance Tuning Guide* for more information on setting this parameter

MAX_ENABLED_ROLES

Property	Description
Parameter type	Integer
Default value	30
Modifiable	No
Range of values	0 to 148
Basic	No

Note: This parameter is deprecated. MAX_ENABLED_ROLES is retained for backward compatibility only.

MAX_ENABLED_ROLES specifies the maximum number of database roles that users can enable, including roles contained within other roles. The actual number of roles that

users can enable is 2 plus the value of `MAX_ENABLED_ROLES`, because each user has two additional roles, `PUBLIC` and the user's own role.

For example, if `MAX_ENABLED_ROLES` is set to 5, then user `HR` can have seven roles enabled (the five enabled by `MAX_ENABLED_ROLES`, plus `PUBLIC` and `HR`).

MAX_SHARED_SERVERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If <code>MAX_SHARED_SERVERS</code> is specified, then it should be greater than or equal to <code>SHARED_SERVERS</code> and less than <code>PROCESSES</code> .
Basic	No

`MAX_SHARED_SERVERS` specifies the maximum number of shared server processes allowed to be running simultaneously. Setting this parameter enables you to reserve process slots for other processes, such as dedicated servers.

When you want to reduce the range of shared servers, you can reduce `MAX_SHARED_SERVERS` before reducing `SHARED_SERVERS`. If `MAX_SHARED_SERVERS` is lower than `SHARED_SERVERS`, then the number of shared servers will not vary but will remain at the constant level specified by `SHARED_SERVERS`. If `MAX_SHARED_SERVERS` is not specified, then a shared server process may be spawned as long as the number of free process slots is greater than 1 / 8 the maximum number of processes, or 2 if `PROCESSES` is less than 24.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Database Concepts* for information on processes
- Your operating system-specific Oracle documentation for the default value and range of values

NLS_CALENDAR

Property	Description
Parameter type	String
Syntax	<code>NLS_CALENDAR = "calendar_system"</code>
Default value	None
Modifiable	ALTER SESSION
Range of values	Any valid calendar format name
Basic	No

`NLS_CALENDAR` specifies which calendar system Oracle uses. It can have one of the following values:

- Arabic Hijrah
- English Hijrah
- Gregorian
- Japanese Imperial
- Persian
- ROC Official (Republic of China)
- Thai Buddha

For example, suppose NLS_CALENDAR is set to "Japanese Imperial", the date format is "E YY-MM-DD". ("E" is the date format element for the abbreviated era name.) If the date is May 15, 1997, then the SYSDATE is displayed as follows:

```
SELECT SYSDATE FROM DUAL;
SYSDATE
-----
H 09-05-15
```

See Also: *Oracle Database Globalization Support Guide* for a listing of available calendar systems

NLS_COMP

Property	Description
Parameter type	String
Syntax	NLS_COMP = { BINARY ANSI }
Default value	BINARY
Modifiable	ALTER SESSION
Basic	No

Normally, comparisons in the WHERE clause and in PL/SQL blocks is binary unless you specify the NLSSORT function. By setting NLS_COMP to ANSI, you indicate that comparisons in the WHERE clause and in PL/SQL blocks should use the linguistic sort specified in the NLS_SORT parameter. You must also define an index on the column for which you want linguistic sorts.

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_CURRENCY

Property	Description
Parameter type	String
Syntax	NLS_CURRENCY = <i>currency_symbol</i>
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid character string, with a maximum of 10 bytes (not including null)

Property	Description
Basic	No

NLS_CURRENCY specifies the string to use as the local currency symbol for the L number format element. The default value of this parameter is determined by NLS_TERRITORY.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Reference* for information on number format elements

NLS_DATE_FORMAT

Property	Description
Parameter type	String
Syntax	NLS_DATE_FORMAT = <i>"format"</i>
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid date format mask but not exceeding a fixed length
Basic	No

NLS_DATE_FORMAT specifies the default date format to use with the TO_CHAR and TO_DATE functions. The default value of this parameter is determined by NLS_TERRITORY.

The value of this parameter can be any valid date format mask, and the value must be surrounded by double quotation marks. For example:

```
NLS_DATE_FORMAT = "MM/DD/YYYY"
```

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DATE_LANGUAGE

Property	Description
Parameter type	String
Syntax	NLS_DATE_LANGUAGE = <i>language</i>
Default value	Derived from NLS_LANGUAGE
Modifiable	ALTER SESSION

Property	Description
Range of values	Any valid NLS_LANGUAGE value

NLS_DATE_LANGUAGE specifies the language to use for the spelling of day and month names and date abbreviations (a.m., p.m., AD, BC) returned by the TO_DATE and TO_CHAR functions.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Reference* for information on the TO_DATE and TO_CHAR functions.
- *Oracle Database Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DUAL_CURRENCY

Property	Description
Parameter type	String
Syntax	NLS_DUAL_CURRENCY = <i>currency_symbol</i>
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid format name up to 10 characters

NLS_DUAL_CURRENCY specifies the dual currency symbol (such as "Euro") for the territory. The default is the dual currency symbol defined in the territory of your current language environment.

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_ISO_CURRENCY

Property	Description
Parameter type	String
Syntax	NLS_ISO_CURRENCY = <i>territory</i>
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid NLS_TERRITORY value

NLS_ISO_CURRENCY specifies the string to use as the international currency symbol for the C number format element.

Local currency symbols can be ambiguous. For example, a dollar sign (\$) can refer to U.S. dollars or Australian dollars. ISO Specification 4217 1987-07-15 defines unique "international" currency symbols for the currencies of specific territories or countries.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Reference* for information on number format elements

NLS_LANGUAGE

Property	Description
Parameter type	String
Syntax	NLS_LANGUAGE = <i>language</i>
Default value	Operating system-dependent, derived from the NLS_LANG environment variable
Modifiable	ALTER SESSION
Range of values	Any valid language name
Basic	Yes

NLS_LANGUAGE specifies the default language of the database. This language is used for messages, day and month names, symbols for AD, BC, a.m., and p.m., and the default sorting mechanism. This parameter also determines the default values of the parameters NLS_DATE_LANGUAGE and NLS_SORT.

See Also:

- *Oracle Database Globalization Support Guide* for a complete list of languages and additional information on this parameter
- Your operating system-specific Oracle documentation and the release notes for your country

NLS_LENGTH_SEMANTICS

Property	Description
Parameter type	String
Syntax	NLS_LENGTH_SEMANTICS = <i>string</i> Example: NLS_LENGTH_SEMANTICS = 'CHAR'
Default value	BYTE
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	BYTE CHAR

NLS_LENGTH_SEMANTICS enables you to create CHAR and VARCHAR2 columns using either byte or character length semantics. Existing columns are not affected.

NCHAR, NVARCHAR2, CLOB, and NCLOB columns are always character-based. You may be required to use byte semantics in order to maintain compatibility with existing applications.

NLS_LENGTH_SEMANTICS does not apply to tables in SYS and SYSTEM. The data dictionary always uses byte semantics.

NLS_NCHAR_CONV_EXCP

Property	Description
Parameter type	String
Syntax	NLS_NCHAR_CONV_EXCP = { TRUE FALSE }
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM

NLS_NCHAR_CONV_EXCP determines whether an error is reported when there is data loss during an implicit or explicit character type conversion between NCHAR/NVARCHAR and CHAR/VARCHAR2. The default value results in no error being reported.

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_NUMERIC_CHARACTERS

Property	Description
Parameter type	String
Syntax	NLS_NUMERIC_CHARACTERS = "decimal_character group_separator"
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION

NLS_NUMERIC_CHARACTERS specifies the characters to use as the group separator and decimal character. It overrides those characters defined implicitly by NLS_TERRITORY. The group separator separates integer groups (that is, thousands, millions, billions, and so on). The decimal separates the integer portion of a number from the decimal portion.

You can specify any character as the decimal or group separator. The two characters specified must be single-byte and must be different from each other. The characters cannot be any numeric character or any of the following characters: plus (+), minus sign (-), less than sign (<), greater than sign (>). Either character can be a space.

For example, if you wish to specify a comma as the decimal character and a space as the group separator, you would set this parameter as follows:

```
NLS_NUMERIC_CHARACTERS = ', '
```

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_SORT

Property	Description
Parameter type	String
Syntax	NLS_SORT = { BINARY <i>linguistic_definition</i> }
Default value	Derived from NLS_LANGUAGE
Modifiable	ALTER SESSION
Range of values	BINARY or any valid linguistic definition name

NLS_SORT specifies the collating sequence for ORDER BY queries.

- If the value is BINARY, then the collating sequence for ORDER BY queries is based on the numeric value of characters (a binary sort that requires less system overhead).
- If the value is a named linguistic sort, sorting is based on the order of the defined linguistic sort. Most (but not all) languages supported by the NLS_LANGUAGE parameter also support a linguistic sort with the same name.

Note: Setting NLS_SORT to anything other than BINARY causes a sort to use a full table scan, regardless of the path chosen by the optimizer. BINARY is the exception because indexes are built according to a binary order of keys. Thus the optimizer can use an index to satisfy the ORDER BY clause when NLS_SORT is set to BINARY. If NLS_SORT is set to any linguistic sort, the optimizer must include a full table scan and a full sort in the execution plan.

You must use the NLS_SORT operator with comparison operations if you want the linguistic sort behavior.

See Also:

- *Oracle Database Globalization Support Guide* for a current listing of values you can specify for this parameter
- *Oracle Database Concepts* for more information on this parameter

NLS_TERRITORY

Property	Description
Parameter type	String
Syntax	NLS_TERRITORY = <i>territory</i>
Default value	Operating system-dependent
Modifiable	ALTER SESSION
Range of values	Any valid territory name
Basic	Yes

NLS_TERRITORY specifies the name of the territory whose conventions are to be followed for day and week numbering.

This parameter also establishes the default date format, the default decimal character and group separator, and the default ISO and local currency symbols.

For information on these settings, see ["NLS_DATE_FORMAT"](#) on page 1-87, ["NLS_NUMERIC_CHARACTERS"](#) on page 1-90, ["NLS_CURRENCY"](#) on page 1-86, and ["NLS_ISO_CURRENCY"](#) on page 1-88.

See Also:

- *Oracle Database Globalization Support Guide* for a complete list of territories
- Your operating system-specific Oracle documentation for the territory-dependent default values for these parameters

NLS_TIMESTAMP_FORMAT

Property	Description
Parameter type	String
Syntax	NLS_TIMESTAMP_FORMAT = " <i>format</i> "
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid datetime format mask

NLS_TIMESTAMP_FORMAT defines the default timestamp format to use with the TO_CHAR and TO_TIMESTAMP functions.

The value must be surrounded by quotation marks as follows:

```
NLS_TIMESTAMP_FORMAT = 'YYYY-MM-DD HH:MI:SS.FF'
```

You can specify the value of NLS_TIMESTAMP_FORMAT by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of NLS_TIMESTAMP_FORMAT by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the ALTER SESSION SET statement.

NLS_TIMESTAMP_TZ_FORMAT

Property	Description
Parameter type	String
Syntax	NLS_TIMESTAMP_TZ_FORMAT = " <i>format</i> "
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid datetime format mask

NLS_TIMESTAMP_TZ_FORMAT defines the default timestamp with time zone format to use with the TO_CHAR and TO_TIMESTAMP_TZ functions.

The value must be surrounded by quotation marks as follows:


```
NLS_TIMESTAMP_TZ_FORMAT = 'YYYY-MM-DD HH:MI:SS.FF TZH:TZM'
```

You can specify the value of `NLS_TIMESTAMP_TZ_FORMAT` by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of `NLS_TIMESTAMP_TZ_FORMAT` by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the `ALTER SESSION SET` statement.

07_DICTIONARY_ACCESSIBILITY

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

`07_DICTIONARY_ACCESSIBILITY` controls restrictions on `SYSTEM` privileges. If the parameter is set to `true`, access to objects in the `SYS` schema is allowed (Oracle7 behavior). The default setting of `false` ensures that system privileges that allow access to objects in "any schema" do not allow access to objects in the `SYS` schema.

For example, if `07_DICTIONARY_ACCESSIBILITY` is set to `false`, then the `SELECT ANY TABLE` privilege allows access to views or tables in any schema except the `SYS` schema (data dictionary tables cannot be accessed). The system privilege `EXECUTE ANY PROCEDURE` allows access on the procedures in any schema except the `SYS` schema.

If this parameter is set to `false` and you need to access objects in the `SYS` schema, then you must be granted explicit object privileges. The following roles, which can be granted to the database administrator, also allow access to dictionary objects:

- `SELECT_CATALOG_ROLE`
- `EXECUTE_CATALOG_ROLE`
- `DELETE_CATALOG_ROLE`

See Also:

- *Oracle Database Upgrade Guide* for more information on this parameter and the roles mentioned here
- *Oracle Database SQL Reference* for information on granting roles

OBJECT_CACHE_MAX_SIZE_PERCENT

Property	Description
Parameter type	Integer
Default value	10
Modifiable	<code>ALTER SESSION, ALTER SYSTEM ... DEFERRED</code>
Range of values	0 to operating system-dependent maximum

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server. `OBJECT_CACHE_MAX_SIZE_PERCENT` specifies the percentage of the optimal cache size that the session object cache can grow past the optimal size. The maximum size is equal to the optimal size plus the product of this percentage and the optimal size. When the cache size exceeds this maximum size, the system will attempt to shrink the cache to the optimal size.

See Also:

- "[OBJECT_CACHE_OPTIMAL_SIZE](#)" on page 1-94 for a description of the object cache
- *Oracle Database Concepts, Pro*C/C++ Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OBJECT_CACHE_OPTIMAL_SIZE

Property	Description
Parameter type	Integer
Default value	102400 (100K)
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	10 KB to operating system-dependent maximum

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server. `OBJECT_CACHE_OPTIMAL_SIZE` specifies (in bytes) the size to which the session object cache is reduced when the size of the cache exceeds the maximum size.

See Also: *Oracle Database Concepts, Pro*C/C++ Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OLAP_PAGE_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>OLAP_PAGE_POOL_SIZE = integer [K M G]</code>
Default value	32 MB
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	32 MB to 2 GB

`OLAP_PAGE_POOL_SIZE` specifies (in bytes) the size of the OLAP pool.

OPEN_CURSORS

Property	Description
Parameter type	Integer
Default value	50
Modifiable	ALTER SYSTEM
Range of values	1 to 4294967295 (4 GB -1)
Basic	Yes

`OPEN_CURSORS` specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. You can use this parameter to prevent a session from opening an excessive number of cursors.

It is important to set the value of `OPEN_CURSORS` high enough to prevent your application from running out of open cursors. The number will vary from one application to another. Assuming that a session does not open the number of cursors specified by `OPEN_CURSORS`, there is no added overhead to setting this value higher than actually needed.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the range of values

OPEN_LINKS

Property	Description
Parameter type	Integer
Default value	4
Modifiable	No
Range of values	0 to 255

`OPEN_LINKS` specifies the maximum number of concurrent open connections to remote databases in one session. These connections include database links, as well as external procedures and cartridges, each of which uses a separate process.

Oracle counts one open link for the following:

- For each user that references a public or private database link
- For each external procedure or cartridge connection when it is executed for the first time

Both types of connections close when the session ends. You can also close a database link connection explicitly by issuing an `ALTER SESSION CLOSE DATABASE LINK` statement.

You should set this parameter to allow for the external procedure and cartridge connections expected during the session plus the number of databases referred to in typical distributed transactions (that is, a single SQL statement that references multiple

databases), so that all the databases can be open to execute the statement. For example, if queries alternately access databases A, B, and C, and `OPEN_LINKS` is set to 2, time will be lost waiting while one connection is broken and another made. Increase the value if many different databases are accessed over time.

This parameter refers only to connections used for distributed transactions. Direct connections to a remote database specified as an application connects are not counted.

If you set `OPEN_LINKS` to 0, then no distributed transactions are allowed.

See Also: "[OPEN_LINKS_PER_INSTANCE](#)" on page 1-96 for information on setting open connections globally for a database instance

OPEN_LINKS_PER_INSTANCE

Property	Description
Parameter type	Integer
Default value	4
Modifiable	No
Range of values	0 to 4294967295 (4 GB -1)
Real Application Clusters	Multiple instances can have different values.

`OPEN_LINKS_PER_INSTANCE` specifies the maximum number of migratable open connections globally for each database instance. XA transactions use migratable open connections so that the connections are cached after a transaction is committed. Another transaction can use the connection, provided the user who created the connection is the same as the user who owns the transaction.

`OPEN_LINKS_PER_INSTANCE` is different from `OPEN_LINKS`, which indicates the number of connections from a session. The `OPEN_LINKS` parameter is not applicable to XA applications.

See Also: "[OPEN_LINKS](#)" on page 1-95

OPTIMIZER_DYNAMIC_SAMPLING

Property	Description
Parameter type	Integer
Default value	If <code>OPTIMIZER_FEATURES_ENABLE</code> is set to 10.0.0 or higher, then 2 If <code>OPTIMIZER_FEATURES_ENABLE</code> is set to 9.2.0, then 1 If <code>OPTIMIZER_FEATURES_ENABLE</code> is set to 9.0.1 or lower, then 0
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to 10

`OPTIMIZER_DYNAMIC_SAMPLING` controls the level of dynamic sampling performed by the optimizer.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

OPTIMIZER_FEATURES_ENABLE

Property	Description
Parameter type	String
Syntax	OPTIMIZER_FEATURES_ENABLE = { 8.0.0 8.0.3 8.0.4 8.0.5 8.0.6 8.0.7 8.1.0 8.1.3 8.1.4 8.1.5 8.1.6 8.1.7 9.0.0 9.0.1 9.2.0 10.0.0 }
Default value	10.0.0
Modifiable	ALTER SESSION, ALTER SYSTEM

OPTIMIZER_FEATURES_ENABLE acts as an umbrella parameter for enabling a series of optimizer features based on an Oracle release number.

For example, if you upgrade your database from release 8.0.6 to release 9.2, but you want to keep the release 8.0.6 optimizer behavior, you can do so by setting this parameter to 8.0.6. At a later time, you can try the enhancements introduced in releases up to and including release 9.2 by setting the parameter to 9.2.0.

[Table 1–2](#) describes some of the optimizer features that are enabled when you set the OPTIMIZER_FEATURES_ENABLE parameter to an 8.0 release or an 8.1 release.

[Table 1–3](#) describes some of the optimizer features that are enabled when you set the OPTIMIZER_FEATURES_ENABLE parameter to a 9.0 release or a 9.2 release.

See Also: *Oracle Database Performance Tuning Guide* for more information about the optimizer and for information about the features listed in the following tables

Table 1–2 Release 8.0 and Release 8.1 Optimizer Features

Features	Release											
	8.0.0	8.0.3	8.0.4	8.0.5	8.0.6	8.0.7	8.1.0	8.1.3	8.1.4	8.1.5	8.1.6	8.1.7
Index fast full scan			X	X	X	X	X	X	X	X	X	X
Consideration of bitmap access paths for tables with only B-tree indexes			X	X	X	X						
Complex view merging			X	X	X	X						
Push-join predicate			X	X	X	X						
Ordered nested loop costing			X	X	X	X						
Improved outer join cardinality calculation					X	X	X	X	X	X	X	X
Improved verification of NULLs inclusion in B-tree indexes										X	X	X
Random distribution method for left of nested loops											X	X
Type-dependent selectivity estimates											X	X

Table 1–2 (Cont.) Release 8.0 and Release 8.1 Optimizer Features

Features	Release											
	8.0.0	8.0.3	8.0.4	8.0.5	8.0.6	8.0.7	8.1.0	8.1.3	8.1.4	8.1.5	8.1.6	8.1.7
Setting of optimizer mode for user recursive SQL											X	X
Improved average row length calculation											X	X
Partition pruning based on subquery predicates											X	X
Common subexpression elimination												X
Use statistics of a column imbedded in some selected functions such as TO_CHAR to compute selectivity												X
Improved partition statistics aggregation												X

Table 1–3 Release 9.0 and Release 9.2 Optimizer Features

Features	Release		
	9.0.0	9.0.1	9.2.0
Index fast full scan	X	X	X
Consideration of bitmap access paths for tables with only B-tree indexes	X	X	X
Complex view merging	X	X	X
Push-join predicate	X	X	X
Ordered nested loop costing	X	X	X
Improved outer join cardinality calculation	X	X	X
Improved verification of NULLs inclusion in B-tree indexes	X	X	X
Random distribution method for left of nested loops	X	X	X
Type-dependent selectivity estimates	X	X	X
Setting of optimizer mode for user recursive SQL	X	X	X
Improved average row length calculation	X	X	X
Partition pruning based on subquery predicates	X	X	X
Common subexpression elimination	X	X	X
Use statistics of a column imbedded in some selected functions such as TO_CHAR to compute selectivity	X	X	X
Improved partition statistics aggregation	X	X	X
Peeking at user-defined bind variables	X	X	X
Index joins	X	X	X
Subquery unnesting	X	X	X

OPTIMIZER_INDEX_CACHING

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to 100

OPTIMIZER_INDEX_CACHING lets you adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

The cost of executing an index using an IN-list iterator or of executing a nested loops join when an index is used to access the inner table depends on the caching of that index in the buffer cache. The amount of caching depends on factors that the optimizer cannot predict, such as the load on the system and the block access patterns of different users.

You can modify the optimizer's assumptions about index caching for nested loops joins and IN-list iterators by setting this parameter to a value between 0 and 100 to indicate the percentage of the index blocks the optimizer should assume are in the cache. Setting this parameter to a higher value makes nested loops joins and IN-list iterators look less expensive to the optimizer. As a result, it will be more likely to pick nested loops joins over hash or sort-merge joins and to pick indexes using IN-list iterators over other indexes or full table scans. The default for this parameter is 0, which results in default optimizer behavior.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

OPTIMIZER_INDEX_COST_ADJ

Property	Description
Parameter type	Integer
Default value	100
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	1 to 10000

OPTIMIZER_INDEX_COST_ADJ lets you tune optimizer behavior for access path selection to be more or less index friendly—that is, to make the optimizer more or less prone to selecting an index access path over a full table scan.

The default for this parameter is 100 percent, at which the optimizer evaluates index access paths at the regular cost. Any other value makes the optimizer evaluate the access path at that percentage of the regular cost. For example, a setting of 50 makes the index access path look half as expensive as normal.

Note: The adjustment does not apply to user-defined cost functions for domain indexes.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter, and on its relationship to "OPTIMIZER_INDEX_CACHING"

OPTIMIZER_MODE

Property	Description
Parameter type	String
Syntax	OPTIMIZER_MODE = { first_rows_[1 10 100 1000] first_rows all_rows }
Default value	all_rows
Modifiable	ALTER SESSION, ALTER SYSTEM

OPTIMIZER_MODE establishes the default behavior for choosing an optimization approach for the instance.

Values:

- `first_rows_n`
The optimizer uses a cost-based approach and optimizes with a goal of best response time to return the first *n* rows (where *n* = 1, 10, 100, 1000).
- `first_rows`
The optimizer uses a mix of costs and heuristics to find a best plan for fast delivery of the first few rows.
- `all_rows`
The optimizer uses a cost-based approach for all SQL statements in the session and optimizes with a goal of best throughput (minimum resource use to complete the entire statement).

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Concepts* and *Oracle Database Performance Tuning Guide* for more information about the optimizer

OS_AUTHENT_PREFIX

Property	Description
Parameter type	String
Syntax	OS_AUTHENT_PREFIX = <i>authentication_prefix</i>
Default value	OPS\$
Modifiable	No

OS_AUTHENT_PREFIX specifies a prefix that Oracle uses to authenticate users attempting to connect to the server. Oracle concatenates the value of this parameter to

the beginning of the user's operating system account name and password. When a connection request is attempted, Oracle compares the prefixed username with Oracle usernames in the database.

The default value of this parameter is `OPS$` for backward compatibility with previous versions. However, you might prefer to set the prefix value to `''` (a null string), thereby eliminating the addition of any prefix to operating system account names.

Note: The text of the `OS_AUTHENT_PREFIX` parameter is case sensitive on some operating systems.

See Also:

- *Oracle Advanced Security Administrator's Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value

OS_ROLES

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

`OS_ROLES` determines whether Oracle or the operating system identifies and manages the roles of each username.

Values:

- TRUE

The operating system completely manages the role grants for all database usernames. When a user attempts to create a session, the username's security domain is initialized using the roles identified by the operating system. A user can subsequently enable as many roles identified by the operating system as specified by the parameter `MAX_ENABLED_ROLES`.

Revocation by Oracle of roles granted by the operating system is ignored, as are any roles previously granted by Oracle.

- FALSE

Oracle identifies and manages the roles.

See Also:

- *Oracle Database Administrator's Guide* and *Oracle Advanced Security Administrator's Guide* for more information on roles and on setting this parameter
- ["REMOTE_OS_ROLES"](#) on page 1-118

PARALLEL_ADAPTIVE_MULTI_USER

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Range of values	true false

PARALLEL_ADAPTIVE_MULTI_USER, when set to `true`, enables an adaptive algorithm designed to improve performance in multiuser environments that use parallel execution. The algorithm automatically reduces the requested degree of parallelism based on the system load at query startup time. The effective degree of parallelism is based on the default degree of parallelism, or the degree from the table or hints, divided by a reduction factor.

The algorithm assumes the following:

- The system has been tuned for optimal performance in a single-user environment
- The parameter PARALLEL_AUTOMATIC_TUNING is set to TRUE

Tables and hints use the default degree of parallelism.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution and on the algorithm described here

PARALLEL_AUTOMATIC_TUNING

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

Note: This parameter is deprecated. PARALLEL_AUTOMATIC_TUNING is retained for backward compatibility only.

When PARALLEL_AUTOMATIC_TUNING is set to `true`, Oracle determines the default values for parameters that control parallel execution. In addition to setting this parameter, you must specify the PARALLEL clause for the target tables in the system. Oracle then tunes all subsequent parallel operations automatically.

If you used parallel execution in a previous release and are now enabling PARALLEL_AUTOMATIC_TUNING, then you should reduce the amount of memory allocated from the shared pool to account for the decreased demand on that pool. This memory will now be allocated from the large pool, and will be computed automatically if LARGE_POOL_SIZE is not specified.

As part of the automatic tuning, Oracle will enable the `PARALLEL_ADAPTIVE_MULTI_USER` parameter. You can override any of the system-provided defaults if desired.

PARALLEL_EXECUTION_MESSAGE_SIZE

Property	Description
Parameter type	Integer
Default value	Operating system-dependent
Modifiable	No
Range of values	2148 to 65535 (64 KB - 1)
Real Application Clusters	Multiple instances must have the same value.

`PARALLEL_EXECUTION_MESSAGE_SIZE` specifies the size of messages for parallel execution (formerly referred to as parallel query, PDML, Parallel Recovery, replication).

On most platforms, the default value is 2148 bytes if `PARALLEL_AUTOMATIC_TUNING` is set to `false`, and 4096 bytes if `PARALLEL_AUTOMATIC_TUNING` is set to `true`. The default value is adequate for most applications. Larger values require a larger shared pool. Larger values result in better performance at the cost of higher memory use. For this reason, replication gets no benefit from increasing the size.

Note: When `PARALLEL_AUTOMATIC_TUNING` is set to `TRUE`, message buffers are allocated out of the large pool. In this case, the default is generally higher.

PARALLEL_INSTANCE_GROUP

Property	Description
Parameter type	String
Syntax	<code>PARALLEL_INSTANCE_GROUP = group_name</code>
Default value	A group consisting of all instances currently active
Modifiable	<code>ALTER SESSION</code> , <code>ALTER SYSTEM</code>
Range of values	Any group name specified in the <code>INSTANCE_GROUPS</code> parameter of any active instance
Real Application Clusters	Multiple instances can have different values.

`PARALLEL_INSTANCE_GROUP` is a Real Application Clusters parameter that you can specify in parallel mode only. Used in conjunction with the `INSTANCE_GROUPS` parameter, it lets you restrict parallel query operations to a limited number of instances.

This parameter identifies the parallel instance group Oracle will use for spawning parallel execution processes. Parallel operations will spawn parallel execution

processes only on instances that specify a matching group in their `INSTANCE_GROUPS` parameter.

If the value of `PARALLEL_INSTANCE_GROUP` does not correspond to an instance group name specified for an active instance, Oracle returns an error.

See Also: *Oracle Real Application Clusters Administrator's Guide* for more information on parallel query execution in a Real Application Clusters environment

PARALLEL_MAX_SERVERS

Property	Description
Parameter type	Integer
Default value	Derived from the values of <code>CPU_COUNT</code> , <code>PARALLEL_AUTOMATIC_TUNING</code> , and <code>PARALLEL_ADAPTIVE_MULTI_USER</code>
Modifiable	<code>ALTER SYSTEM</code>
Range of values	0 to 3599
Real Application Clusters	Multiple instances must have the same value.

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

`PARALLEL_MAX_SERVERS` specifies the maximum number of parallel execution processes and parallel recovery processes for an instance. As demand increases, Oracle increases the number of processes from the number created at instance startup up to this value.

If you set this parameter too low, some queries may not have a parallel execution process available to them during query processing. If you set it too high, memory resource shortages may occur during peak periods, which can degrade performance.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PARALLEL_MIN_PERCENT

Property	Description
Parameter type	Integer
Default value	0
Modifiable	<code>ALTER SESSION</code>
Range of values	0 to 100
Real Application Clusters	Multiple instances can have different values.

`PARALLEL_MIN_PERCENT` operates in conjunction with `PARALLEL_MAX_SERVERS` and `PARALLEL_MIN_SERVERS`. It lets you specify the minimum percentage of parallel

execution processes (of the value of `PARALLEL_MAX_SERVERS`) required for parallel execution. Setting this parameter ensures that parallel operations will not execute sequentially unless adequate resources are available. The default value of 0 means that no minimum percentage of processes has been set.

Consider the following settings:

```
PARALLEL_MIN_PERCENT = 50
PARALLEL_MIN_SERVERS = 5
PARALLEL_MAX_SERVERS = 10
```

If 8 of the 10 parallel execution processes are busy, only 2 processes are available. If you then request a query with a degree of parallelism of 8, the minimum 50% will not be met.

You can use this parameter in conjunction with `PARALLEL_ADAPTIVE_MULTI_USER`. In a multi-user environment, an individual user or application can set `PARALLEL_MIN_PERCENT` to a minimum value until sufficient resources are available on the system and an acceptable degree of parallelism is returned.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution
- "[PARALLEL_MAX_SERVERS](#)" on page 1-104, "[PARALLEL_MIN_SERVERS](#)" on page 1-105, and "[PARALLEL_ADAPTIVE_MULTI_USER](#)" on page 1-102

PARALLEL_MIN_SERVERS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to value of <code>PARALLEL_MAX_SERVERS</code>
Real Application Clusters	Multiple instances can have different values.

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

`PARALLEL_MIN_SERVERS` specifies the minimum number of parallel execution processes for the instance. This value is the number of parallel execution processes Oracle creates when the instance is started.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PARALLEL_THREADS_PER_CPU

Property	Description
Parameter type	Integer
Default value	Operating system-dependent, usually 2
Modifiable	ALTER SYSTEM
Range of values	Any nonzero number

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

PARALLEL_THREADS_PER_CPU specifies the default degree of parallelism for the instance and determines the parallel adaptive and load balancing algorithms. The parameter describes the number of parallel execution processes or **threads** that a CPU can handle during parallel execution.

The default is platform-dependent and is adequate in most cases. You should decrease the value of this parameter if the machine appears to be overloaded when a representative parallel query is executed. You should increase the value if the system is I/O bound.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PGA_AGGREGATE_TARGET

Property	Description
Parameter type	Big integer
Syntax	PGA_AGGREGATE_TARGET = <i>integer</i> [K M G]
Default value	10 MB or 20% of the size of the SGA, whichever is greater
Modifiable	ALTER SYSTEM
Range of values	Minimum: 10 MB Maximum: 4096 GB - 1
Basic	Yes

PGA_AGGREGATE_TARGET specifies the target aggregate PGA memory available to all server processes attached to the instance. You must set this parameter to enable the automatic sizing of SQL working areas used by memory-intensive SQL operators such as sort, group-by, hash-join, bitmap merge, and bitmap create.

Oracle uses this parameter as a target for PGA memory. Use this parameter to determine the optimal size of each work area allocated in AUTO mode (in other words, when WORKAREA_SIZE_POLICY is set to AUTO).

Oracle attempts to keep the amount of private memory below the target specified by this parameter by adapting the size of the work areas to private memory. When increasing the value of this parameter, you indirectly increase the memory allotted to

work areas. Consequently, more memory-intensive operations are able to run fully in memory and less will work their way over to disk.

When setting this parameter, you should examine the total memory on your system that is available to the Oracle instance and subtract the SGA. You can assign the remaining memory to `PGA_AGGREGATE_TARGET`.

PLSQL_CODE_TYPE

Property	Description
Parameter type	String
Syntax	<code>PLSQL_CODE_TYPE = { INTERPRETED NATIVE }</code>
Default value	INTERPRETED
Modifiable	ALTER SESSION, ALTER SYSTEM

`PLSQL_CODE_TYPE` specifies the compilation mode for PL/SQL library units.

Values:

- INTERPRETED

PL/SQL library units will be compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine.

- NATIVE

PL/SQL library units (with the possible exception of top-level anonymous PL/SQL blocks) will be compiled to native (machine) code. Such modules will be executed natively without incurring any interpreter overhead.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

If a PL/SQL library unit is compiled native, all subsequent automatic recompilations of that library unit will use native compilation.

PLSQL_COMPILER_FLAGS

Property	Description
Parameter type	String
Syntax	<code>PLSQL_COMPILER_FLAGS = { [DEBUG NON_DEBUG] [INTERPRETED NATIVE] }</code>
Default value	INTERPRETED, NON_DEBUG
Modifiable	ALTER SESSION, ALTER SYSTEM

Note: This parameter is deprecated in favor of the `PLSQL_CODE_TYPE` and `PLSQL_DEBUG` parameters. Oracle recommends that you use `PLSQL_CODE_TYPE` and `PLSQL_DEBUG` instead. `PLSQL_COMPILER_FLAGS` is retained for backward compatibility only.

PLSQL_COMPILER_FLAGS specifies a list of flags for the PL/SQL compiler as a comma-separated list of strings.

Values:

- INTERPRETED
PL/SQL library units will be compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine.
- NATIVE
PL/SQL library units (with the possible exception of top-level anonymous PL/SQL blocks) will be compiled to native (machine) code. Such modules will be executed natively without incurring any interpreter overhead.
- DEBUG
PL/SQL library units will be compiled for debugging
- NON_DEBUG
PL/SQL library units will be compiled for normal execution

The following combinations of flags are invalid specifications for the PLSQL_COMPILER_FLAGS parameter:

- NATIVE & DEBUG
- NATIVE & INTERPRETED
- NON_DEBUG & DEBUG

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with the library unit. If a PL/SQL library unit is compiled native, all subsequent automatic recompilations of that library unit will use native compilation.

PLSQL_DEBUG

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false

PLSQL_DEBUG specifies whether or not PL/SQL library units will be compiled for debugging.

Values:

- true
PL/SQL library units will be compiled for debugging
- false
PL/SQL library units will be compiled for normal execution

When PLSQL_DEBUG is set to true, PL/SQL library units are always compiled INTERPRETED in order to be debuggable.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

PLSQL_NATIVE_LIBRARY_DIR

Property	Description
Parameter type	String
Syntax	PLSQL_NATIVE_LIBRARY_DIR = <i>directory</i>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	Any valid directory path

PLSQL_NATIVE_LIBRARY_DIR is a parameter used by the PL/SQL compiler. It specifies the name of a directory where the shared objects produced by the native compiler are stored.

PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to $2^{32} - 1$ (max value represented by 32 bits)

PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT specifies the number of subdirectories created by the database administrator in the directory specified by PLSQL_NATIVE_LIBRARY_DIR.

When using the PL/SQL native compiler, Performance of file create/open operations is unacceptably slow if the number of files in a directory is very large. It is usually advisable to create subdirectories and use this parameter if the total number of different PL/SQL packages that may need to be compiled natively by all users of the database instance exceeds 10000.

The subdirectories should have names corresponding to zero-based decimal numbers, prefixed by d. For example, the database administrator can create 1000 subdirectories named d0, d1, ... d999, and set PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT to 1000.

PLSQL_OPTIMIZE_LEVEL

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to 2

`PLSQL_OPTIMIZE_LEVEL` specifies the optimization level that will be used to compile PL/SQL library units. The higher the setting of this parameter, the more effort the compiler makes to optimize PL/SQL library units.

Values:

- 0
Maintains the evaluation order and hence the pattern of side effects, exceptions, and package initializations of Oracle9i and earlier releases. Also removes the new semantic identity of `BINARY_INTEGER` and `PLS_INTEGER` and restores the earlier rules for the evaluation of integer expressions. Although code will run somewhat faster than it did in Oracle9i, use of level 0 will forfeit most of the performance gains of PL/SQL in Oracle Database 10g.
- 1
Applies a wide range of optimizations to PL/SQL programs including the elimination of unnecessary computations and exceptions, but generally does not move source code out of its original source order.
- 2
Applies a wide range of modern optimization techniques beyond those of level 1 including changes which may move source code relatively far from its original location.

Generally, setting this parameter to 2 pays off in better execution performance. If, however, the compiler runs slowly on a particular source module or if optimization does not make sense for some reason (for example, during rapid turnaround development), then setting this parameter to 1 will result in almost as good a compilation with less use of compile-time resources.

The value of this parameter is stored persistently with the library unit.

PLSQL_V2_COMPATIBILITY

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false

PL/SQL Version 2 allows some abnormal behavior that Version 8 disallows. If you want to retain that behavior for backward compatibility, set `PLSQL_V2_COMPATIBILITY` to `true`. If you set it to `false`, PL/SQL Version 8 behavior is enforced and Version 2 behavior is not allowed.

See Also: *PL/SQL User's Guide and Reference* for a description of the differences between PL/SQL Version 2 and Version 8, and for more information on setting this parameter

PLSQL_WARNINGS

Property	Description
Parameter type	String
Syntax	<pre>PLSQL_WARNINGS = 'value_clause' [, 'value_clause'] ...</pre> <p>value_clause::=</p> <pre>{ ENABLE DISABLE ERROR } : { ALL SEVERE INFORMATIONAL PERFORMANCE { integer (integer [, integer] ...) } }</pre>
Default value	'DISABLE:ALL'
Modifiable	ALTER SESSION, ALTER SYSTEM
Examples	<pre>PLSQL_WARNINGS = 'ENABLE:SEVERE', 'DISABLE:INFORMATIONAL'; PLSQL_WARNINGS = 'DISABLE:ALL'; PLSQL_WARNINGS = 'DISABLE:5000', 'ENABLE:5001', 'ERROR:5002'; PLSQL_WARNINGS = 'ENABLE:(5000,5001,5002)', 'DISABLE:(6000,6001)';</pre>

PLSQL_WARNINGS enables or disables the reporting of warning messages by the PL/SQL compiler, and specifies which warning messages to show as errors.

value_clause

Multiple value clauses may be specified, enclosed in quotes and separated by commas. Each value clause is composed of a qualifier, a colon (:), and a modifier.

Qualifier values:

- ENABLE
Enable a specific warning or a set of warnings
- DISABLE
Disable a specific warning or a set of warnings
- ERROR
Treat a specific warning or a set of warnings as errors

Modifier values:

- ALL
Apply the qualifier to all warning messages
- SEVERE
Apply the qualifier to only those warning messages in the SEVERE category
- INFORMATIONAL

Apply the qualifier to only those warning messages in the INFORMATIONAL category

- PERFORMANCE

Apply the qualifier to only those warning messages in the PERFORMANCE category

PRE_PAGE_SGA

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

PRE_PAGE_SGA determines whether Oracle reads the entire SGA into memory at instance startup. Operating system page table entries are then prebuilt for each page of the SGA. This setting can increase the amount of time necessary for instance startup, but it is likely to decrease the amount of time necessary for Oracle to reach its full performance capacity after startup.

Note: This setting does not prevent your operating system from paging or swapping the SGA after it is initially read into memory.

PRE_PAGE_SGA can increase the process startup duration, because every process that starts must access every page in the SGA. The cost of this strategy is fixed; however, you might simply determine that 20,000 pages must be touched every time a process starts. This approach can be useful with some applications, but not with all applications. Overhead can be significant if your system frequently creates and destroys processes by, for example, continually logging on and logging off.

The advantage that PRE_PAGE_SGA can afford depends on page size. For example, if the SGA is 80 MB in size and the page size is 4 KB, then 20,000 pages must be touched to refresh the SGA ($80,000/4 = 20,000$).

If the system permits you to set a 4 MB page size, then only 20 pages must be touched to refresh the SGA ($80,000/4,000 = 20$). The page size is operating system-specific and generally cannot be changed. Some operating systems, however, have a special implementation for shared memory whereby you can change the page size.

PROCESSES

Property	Description
Parameter type	Integer
Default value	Derived from PARALLEL_MAX_SERVERS
Modifiable	No
Range of values	6 to operating system-dependent
Basic	Yes

Property	Description
Real Application Clusters	Multiple instances can have different values.

PROCESSES specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value should allow for all background processes such as locks, job queue processes, and parallel execution processes.

The default values of the SESSIONS and TRANSACTIONS parameters are derived from this parameter. Therefore, if you change the value of PROCESSES, you should evaluate whether to adjust the values of those derived parameters.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter in exclusive mode
- *Oracle Real Application Clusters Administrator's Guide* for information on setting this parameter in a Real Application Clusters environment
- Your operating system-specific Oracle documentation for the range of values

QUERY_REWRITE_ENABLED

Property	Description
Parameter type	String
Syntax	QUERY_REWRITE_ENABLED = { false true force }
Default value	If OPTIMIZER_FEATURES_ENABLE is set to 10.0.0 or higher, then true If OPTIMIZER_FEATURES_ENABLE is set to 9.2.0 or lower, then false
Modifiable	ALTER SESSION, ALTER SYSTEM
Real Application Clusters	Multiple instances can have different values.

QUERY_REWRITE_ENABLED allows you to enable or disable query rewriting globally for the database.

Values:

- false
Oracle does not use rewrite.
- true
Oracle costs the query with rewrite and without rewrite and chooses the method with the lower cost.
- force
Oracle always uses rewrite and does not evaluate the cost before doing so. Use force when you know that the query will always benefit from rewrite and when reduction in compile time is important.

To take advantage of query rewrite for a particular materialized view, you must enable query rewrite for that materialized view, and you must enable cost-based optimization.

See Also:

- *Oracle Data Warehousing Guide* for information on query rewrite of materialized views
- *Oracle Database Performance Tuning Guide* and "[OPTIMIZER_MODE](#)" on page 1-100 for information on cost-based optimization

QUERY_REWRITE_INTEGRITY

Property	Description
Parameter type	String
Syntax	QUERY_REWRITE_INTEGRITY = { enforced trusted stale_tolerated }
Default value	enforced
Modifiable	ALTER SESSION, ALTER SYSTEM
Real Application Clusters	Multiple instances can have different values.

QUERY_REWRITE_INTEGRITY determines the degree to which Oracle must enforce query rewriting. At the safest level, Oracle does not use query rewrite transformations that rely on unenforced relationships.

Values:

- enforced
Oracle enforces and guarantees consistency and integrity.
- trusted
Oracle allows rewrites using relationships that have been declared, but that are not enforced by Oracle.
- stale_tolerated
Oracle allows rewrites using unenforced relationships. Materialized views are eligible for rewrite even if they are known to be inconsistent with the underlying detail data.

RDBMS_SERVER_DN

Property	Description
Parameter type	X.500 Distinguished Name
Default value	There is no default value.
Modifiable	No
Range of values	All X.500 Distinguished Name format values

RDBMS_SERVER_DN specifies the Distinguished Name (DN) of the Oracle server. It is used for retrieving Enterprise Roles from an enterprise directory service.

If you do not want to use a directory for enterprise user and privilege management, but prefer to use SSL authentication alone, do not set this parameter.

See Also: *Oracle Advanced Security Administrator's Guide* for more information on enterprise roles and the enterprise directory service

READ_ONLY_OPEN_DELAYED

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

READ_ONLY_OPEN_DELAYED determines when datafiles in read-only tablespaces are accessed.

Values:

- true

The datafiles are accessed for the first time only when an attempt is made to read data stored within them.
- false

The datafiles are accessed at database open time.

You can use this parameter to speed up some operations (primarily opening the database) for very large databases when substantial portions of the database are stored in read-only tablespaces. Consider setting this parameter to `true` for such databases, especially if portions of the read-only data are stored on slow-access devices or hierarchical storage.

See Also: *Oracle Database Administrator's Guide* for information on the consequences of delaying access of datafiles in read-only tablespaces

RECOVERY_PARALLELISM

Property	Description
Parameter type	Integer
Default value	Operating system-dependent
Modifiable	No
Range of values	Operating system-dependent, but cannot exceed PARALLEL_MAX_SERVERS

RECOVERY_PARALLELISM specifies the number of processes to participate in instance or crash recovery. A value of 0 or 1 indicates that recovery is to be performed serially by one process.

See Also:

- *Oracle Database Performance Tuning Guide* for information on setting this parameter in exclusive mode
- *Oracle Real Application Clusters Administrator's Guide* for information on setting this parameter in a Real Application Clusters environment

REMOTE_ARCHIVE_ENABLE

Property	Description
Parameter type	String
Syntax	REMOTE_ARCHIVE_ENABLE = { receive [, send] false true }
Default value	true
Modifiable	No

REMOTE_ARCHIVE_ENABLE enables or disables the sending of redo archival to remote destinations and the receipt of remotely archived redo.

Values:

- `receive`
Disables the sending of redo archival to remote destinations and enables the receipt of remotely archived redo.
- `send`
Enables the sending of redo archival to remote destinations and disables the receipt of remotely archived redo.
- `false`
Disables both the sending of redo archival to remote destinations and the receipt of remotely archived redo.
- `receive, send or true`
Enables both the sending of redo archival to remote destinations and the receipt of remotely archived redo.

REMOTE_DEPENDENCIES_MODE

Property	Description
Parameter type	String
Syntax	REMOTE_DEPENDENCIES_MODE = { TIMESTAMP SIGNATURE }
Default value	TIMESTAMP
Modifiable	ALTER SESSION, ALTER SYSTEM

REMOTE_DEPENDENCIES_MODE specifies how Oracle should handle dependencies upon remote PL/SQL stored procedures.

Values:

- **TIMESTAMP**

The client running the procedure compares the timestamp recorded on the server-side procedure with the current timestamp of the local procedure and executes the procedure only if the timestamps match.

- **SIGNATURE**

Oracle allows the procedure to execute as long as the signatures are considered safe. This setting allows client PL/SQL applications to be run without recompilation.

See Also: *Oracle Database Application Developer's Guide - Fundamentals* for information about the consequences of the settings of this parameter

REMOTE_LISTENER

Property	Description
Parameter type	String
Syntax	REMOTE_LISTENER = <i>network_name</i>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	Yes

REMOTE_LISTENER specifies a network name that resolves to an address or address list of Oracle Net remote listeners (that is, listeners that are not running on the same machine as this instance). The address or address list is specified in the TNSNAMES.ORA file or other address repository as configured for your system.

See Also:

- *Oracle Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle Net Services Administrator's Guide* and your operating system-specific Oracle documentation for more information about specifying network addresses for the protocols on your system

REMOTE_LOGIN_PASSWORDFILE

Property	Description
Parameter type	String
Syntax	REMOTE_LOGIN_PASSWORDFILE= {NONE SHARED EXCLUSIVE}
Default value	NONE
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

REMOTE_LOGIN_PASSWORDFILE specifies whether Oracle checks for a password file and how many databases can use the password file.

Values:

- NONE
Oracle ignores any password file. Therefore, privileged users must be authenticated by the operating system.
- SHARED
More than one database can use a password file. However, the only user recognized by the password file is SYS.
- EXCLUSIVE
The password file can be used by only one database and the password file can contain names other than SYS.

Note: This setting is required for Real Application Clusters

See Also: *Oracle Database Administrator's Guide* for more information about secure connections for privileged users

REMOTE_OS_AUTHENT

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

REMOTE_OS_AUTHENT specifies whether remote clients will be authenticated with the value of the OS_AUTHENT_PREFIX parameter.

See Also:

- *Oracle Advanced Security Administrator's Guide* for more information on setting this parameter
- ["OS_AUTHENT_PREFIX"](#) on page 1-100

REMOTE_OS_ROLES

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

REMOTE_OS_ROLES specifies whether operating system roles are allowed for remote clients. The default value, `false`, causes Oracle to identify and manage roles for remote clients.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- "[OS_ROLES](#)" on page 1-101

REPLICATION_DEPENDENCY_TRACKING

Property	Description
Parameter type	Boolean
Default value	<code>true</code>
Modifiable	No
Range of values	<code>true</code> <code>false</code>

REPLICATION_DEPENDENCY_TRACKING enables or disables dependency tracking for read/write operations to the database. Dependency tracking is essential for propagating changes in a replicated environment in parallel.

Values:

- `TRUE`
Enables dependency tracking.
- `FALSE`
Allows read/write operations to the database to run faster, but does not produce dependency information for Oracle to perform parallel propagation.

Note: Do not specify this value unless you are sure that your application will not perform any read/write operations to the replicated tables.

See Also: *Oracle Database Advanced Replication* for more information on parallel propagation dependency tracking

RESOURCE_LIMIT

Property	Description
Parameter type	Boolean
Default value	<code>false</code>
Modifiable	ALTER SYSTEM
Range of values	<code>true</code> <code>false</code>

RESOURCE_LIMIT determines whether resource limits are enforced in database profiles.

Values:

- TRUE
Enables the enforcement of resource limits
- FALSE
Disables the enforcement of resource limits

See Also: *Oracle Database Administrator's Guide* and *Oracle Database SQL Reference* for more information on setting resource limits for profiles

RESOURCE_MANAGER_PLAN

Property	Description
Parameter type	String
Syntax	RESOURCE_MANAGER_PLAN = <i>plan_name</i>
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	Any valid character string

RESOURCE_MANAGER_PLAN specifies the top-level resource plan to use for an instance. The resource manager will load this top-level plan along with all its descendants (subplans, directives, and consumer groups). If you do not specify this parameter, the resource manager is off by default.

You can change the setting of this parameter using the ALTER SYSTEM statement to turn on the resource manager (if it was previously off) or to turn off the resource manager or change the current plan (if it was previously on). If you specify a plan that does not exist in the data dictionary, Oracle returns an error message.

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans
- *PL/SQL Packages and Types Reference* for information on the DBMS_RESOURCE_MANAGER and DBMS_RESOURCE_MANAGER_PRIVS packages
- "[DBA_RSRC_PLANS](#)" on page 3-114, "[DBA_RSRC_PLAN_DIRECTIVES](#)" on page 3-113, and the various V\$RSRC_* dynamic performance views in [Part III](#) for information on existing resource plans

RESUMABLE_TIMEOUT

Property	Description
Parameter type	Integer
Default value	0 (seconds)
Modifiable	ALTER SESSION, ALTER SYSTEM

Property	Description
Range of values	0 to $2^{31} - 1$ (in seconds)
Real Application Clusters	Multiple instances can have different values.

RESUMABLE_TIMEOUT enables or disables resumable statements and specifies resumable timeout at the system level.

See Also: *Oracle Database Administrator's Guide* for more information about enabling resumable space allocation, what conditions are correctable, and what statements can be made resumable

ROLLBACK_SEGMENTS

Property	Description
Parameter type	String
Syntax	ROLLBACK_SEGMENTS = (segment_name [, segment_name] ...)
Default value	The instance uses public rollback segments by default if you do not specify this parameter
Modifiable	No
Range of values	Any rollback segment names listed in DBA_ROLLBACK_SEGS except SYSTEM
Basic	Yes
Real Application Clusters	Multiple instances must have different values.

ROLLBACK_SEGMENTS allocates one or more rollback segments by name to this instance. If you set this parameter, the instance acquires all of the rollback segments named in this parameter, even if the number of rollback segments exceeds the minimum number required by the instance (calculated as $\text{TRANSACTIONS} / \text{TRANSACTIONS_PER_ROLLBACK_SEGMENT}$).

You cannot change the value of this parameter dynamically, but you can change its value and then restart the instance. Although this parameter usually specifies private rollback segments, it can also specify public rollback segments if they are not already in use.

To find the name, segment ID number, and status of each rollback segment in the database, query the data dictionary view DBA_ROLLBACK_SEGS.

When UNDO_MANAGEMENT is set to AUTO, ROLLBACK_SEGMENTS is ignored.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Real Application Clusters Deployment and Performance Guide* for information on setting this parameter in a Real Application Clusters environment
- "[DBA_ROLLBACK_SEGS](#)" on page 3-111

SERIAL_REUSE

Property	Description
Parameter type	String
Syntax	SERIAL_REUSE = { DISABLE SELECT DML PLSQL ALL }
Default value	DISABLE
Modifiable	No

SERIAL_REUSE specifies which types of cursors make use of the serial-reusable memory feature. This feature allocates private cursor memory in the SGA so that it can be reused (serially, not concurrently) by sessions executing the same cursor.

Values:

- DISABLE
Disables the option for all SQL statement types. This value overrides any other values included in the list.
- SELECT
Enables the option for SELECT statements.
- DML
Enables the option for DML statements.
- PLSQL
Currently has no effect (although PLSQL packages do support the serial-reuse memory option using PLSQL pragmas).
- ALL
Enables the option for both DML and SELECT statements. Equivalent to setting SELECT, DML, and PLSQL.

Note: If `CURSOR_SPACE_FOR_TIME` is set to `TRUE`, then the value of `SERIAL_REUSE` is ignored and treated as if it were set to `DISABLE`.

See Also: "[CURSOR_SPACE_FOR_TIME](#)" on page 1-32

SERVICE_NAMES

Property	Description
Parameter type	String
Syntax	<code>SERVICE_NAMES = db_service_name [, db_service_name [...]]</code>
Default value	<code>DB_NAME.DB_DOMAIN</code> if defined
Modifiable	<code>ALTER SYSTEM</code>
Range of values	Any ASCII string or comma-separated list of string names
Basic	No
Real Application Clusters	You must set this parameter for every instance.

`SERVICE_NAMES` specifies one or more names by which clients can connect to the instance. The instance registers its service names with the listener. When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

You can specify multiple service names in order to distinguish among different uses of the same database. For example:

```
SERVICE_NAMES = sales.acme.com, widgetsales.acme.com
```

You can also use service names to identify a single service that is available from two different databases through the use of replication.

If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the `DB_DOMAIN` parameter. If `DB_DOMAIN` is not specified, then no domain will be applied to the non-qualified `SERVICE_NAMES` values.

See Also:

- *Oracle Net Services Administrator's Guide* for more information on this parameter and its settings
- "[DB_DOMAIN](#)" on page 1-38

SESSION_CACHED_CURSORS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	<code>ALTER SESSION</code>
Range of values	0 to operating system-dependent
Real Application Clusters	Multiple instances can have different values.

`SESSION_CACHED_CURSORS` specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find

the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed.

This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are re-executed by a user.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter and its uses

SESSION_MAX_OPEN_FILES

Property	Description
Parameter type	Integer
Default value	10
Modifiable	No
Range of values	1 to either 50 or the value of MAX_OPEN_FILES defined at the operating system level, whichever is less

SESSION_MAX_OPEN_FILES specifies the maximum number of BFILEs that can be opened in any session. Once this number is reached, subsequent attempts to open more files in the session by using DBMS_LOB.FILEOPEN() or OCILobFileOpen() will fail. The maximum value for this parameter depends on the equivalent parameter defined for the underlying operating system.

See Also:

- *Oracle Database Application Developer's Guide - Large Objects* for information on LOBs in general and BFILEs in particular
- *PL/SQL Packages and Types Reference* for information on the DBMS_LOB.FILEOPEN() procedure
- *Oracle Call Interface Programmer's Guide* for information on the OCILobFileOpen() procedure

SESSIONS

Property	Description
Parameter type	Integer
Default value	Derived: (1.1 * PROCESSES) + 5
Modifiable	No
Range of values	1 to 2 ³¹
Basic	Yes

SESSIONS specifies the maximum number of sessions that can be created in the system. Because every login requires a session, this parameter effectively determines the maximum number of concurrent users in the system. You should always set this parameter explicitly to a value equivalent to your estimate of the maximum number of concurrent users, plus the number of background processes, plus approximately 10% for recursive sessions.

Oracle uses the default value of this parameter as its minimum. Values between 1 and the default do not trigger errors, but Oracle ignores them and uses the default instead.

The default values of the `ENQUEUE_RESOURCES` and `TRANSACTIONS` parameters are derived from `SESSIONS`. Therefore, if you increase the value of `SESSIONS`, you should consider whether to adjust the values of `ENQUEUE_RESOURCES` and `TRANSACTIONS` as well.

In a shared server environment, the value of `PROCESSES` can be quite small. Therefore, Oracle recommends that you adjust the value of `SESSIONS` to approximately 1.1 * total number of *connections*.

See Also:

- *Oracle Database Concepts* for more information on memory structures and processes
- "[ENQUEUE_RESOURCES](#)" on page 1-51 and "[TRANSACTIONS](#)" on page 1-141

SGA_MAX_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>SGA_MAX_SIZE = integer [K M G]</code>
Default value	Initial size of SGA at startup, dependent on the sizes of different pools in the SGA, such as buffer cache, shared pool, large pool, and so on.
Modifiable	No
Range of values	0 to operating system-dependent

`SGA_MAX_SIZE` specifies the maximum size of the SGA for the lifetime of the instance.

SGA_TARGET

Property	Description
Parameter type	Big integer
Syntax	<code>SGA_TARGET = integer [K M G]</code>
Default value	0 (SGA autotuning is disabled)
Modifiable	<code>ALTER SYSTEM</code>
Range of values	64 to operating system-dependent
Basic	Yes

`SGA_TARGET` specifies the total size of all SGA components. If `SGA_TARGET` is specified, then the following memory pools are automatically sized:

- Buffer cache (`DB_CACHE_SIZE`)
- Shared pool (`SHARED_POOL_SIZE`)
- Large pool (`LARGE_POOL_SIZE`)
- Java pool (`JAVA_POOL_SIZE`)

If these automatically tuned memory pools are set to non-zero values, then those values are used as minimum levels by Automatic Shared Memory Management. You would set minimum values if an application component needs a minimum amount of memory to function properly.

The following pools are manually sized components and are not affected by Automatic Shared Memory Management:

- Log buffer
- Other buffer caches, such as `KEEP`, `RECYCLE`, and other block sizes
- Streams pool
- Fixed SGA and other internal allocations

The memory allocated to these pools is deducted from the total available for `SGA_TARGET` when Automatic Shared Memory Management computes the values of the automatically tuned memory pools.

See Also:

- *Oracle Database Concepts* for information on automatic SGA management
- *Oracle Database Administrator's Guide* for information on managing the SGA

SHADOW_CORE_DUMP

Property	Description
Parameter type	String
Syntax	<code>SHADOW_CORE_DUMP = { partial full none }</code>
Default value	<code>partial</code>
Modifiable	No

`SHADOW_CORE_DUMP` specifies whether Oracle includes the SGA in the core file for foreground (client) processes.

Values:

- `partial`
Oracle does not include the SGA in the core dump.
- `full`
Oracle includes the SGA in the core dump.
- `none`
No core files will be generated for foreground processes.

See Also: ["BACKGROUND_CORE_DUMP"](#) on page 1-22

SHARED_MEMORY_ADDRESS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No

SHARED_MEMORY_ADDRESS and HI_SHARED_MEMORY_ADDRESS specify the starting address at runtime of the system global area (SGA). This parameter is ignored on the many platforms that specify the SGA's starting address at linktime.

Use this parameter to specify the entire address on 32-bit platforms and to specify the low-order 32 bits of a 64-bit address on 64-bit platforms. Use HI_SHARED_MEMORY_ADDRESS to specify the high-order 32 bits of a 64-bit address on 64-bit platforms. If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

See Also: "[HI_SHARED_MEMORY_ADDRESS](#)" on page 1-59

SHARED_POOL_RESERVED_SIZE

Property	Description
Parameter type	Big integer
Syntax	SHARED_POOL_RESERVED_SIZE = <i>integer</i> [K M G]
Default value	5% of the value of SHARED_POOL_SIZE
Modifiable	No
Range of values	Minimum: 5000 Maximum: one half of the value of SHARED_POOL_SIZE

SHARED_POOL_RESERVED_SIZE specifies (in bytes) the shared pool space that is reserved for large contiguous requests for shared pool memory.

You can use this parameter to avoid performance degradation in the shared pool in situations where pool fragmentation forces Oracle to search for and free chunks of unused pool to satisfy the current request.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "[SHARED_POOL_SIZE](#)" on page 1-127

SHARED_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	SHARED_POOL_SIZE = <i>integer</i> [K M G]

Property	Description
Default value	<p>If <code>SGA_TARGET</code> is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool.</p> <p>If <code>SGA_TARGET</code> is not set (32-bit platforms): 32 MB, rounded up to the nearest granule size</p> <p>If <code>SGA_TARGET</code> is not set (64-bit platforms): 84 MB, rounded up to the nearest granule size</p>
Modifiable	ALTER SYSTEM
Range of values	<p>Minimum: the granule size</p> <p>Maximum: operating system-dependent</p>

`SHARED_POOL_SIZE` specifies (in bytes) the size of the shared pool. The shared pool contains shared cursors, stored procedures, control structures, and other structures. If you set `PARALLEL_AUTOMATIC_TUNING` to `false`, then Oracle also allocates parallel execution message buffers from the shared pool. Larger values improve performance in multi-user systems. Smaller values use less memory.

You can monitor utilization of the shared pool by querying the view `V$SGASTAT`.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Upgrade Guide* for information on parallel execution message buffers
- "[PARALLEL_AUTOMATIC_TUNING](#)" on page 1-102 and "[V\\$SGASTAT](#)" on page 5-50

SHARED_SERVER_SESSIONS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If <code>SHARED_SERVER_SESSIONS</code> is specified, then it should be less than <code>SESSIONS</code> . If <code>SHARED_SERVER_SESSIONS</code> is not specified, then a shared server session may be created as long as there is a free session slot.

`SHARED_SERVER_SESSIONS` specifies the total number of shared server sessions to allow. Setting this parameter enables you to reserve user sessions for dedicated servers.

See Also: *Oracle Database Concepts* for more information on sessions

SHARED_SERVERS

Property	Description
Parameter type	Integer
Default value	0, meaning that shared server is not on. If you are using shared server architecture or if the DISPATCHERS parameter is set such that the total number of dispatchers is more than 0, then the default value is 1.
Modifiable	ALTER SYSTEM
Range of values	The value of this parameter should be less than MAX_SHARED_SERVERS. If it is greater than or equal to MAX_SHARED_SERVERS, then the number of servers will not be self-tuned but will remain constant, as specified by SHARED_SERVERS.
Basic	Yes

SHARED_SERVERS specifies the number of server processes that you want to create when an instance is started. If system load decreases, then this minimum number of servers is maintained. Therefore, you should take care not to set SHARED_SERVERS too high at system startup.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter

SKIP_UNUSABLE_INDEXES

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true false

SKIP_UNUSABLE_INDEXES enables or disables the use and reporting of tables with unusable indexes or index partitions.

Values:

- true

Disables error reporting of indexes and index partitions marked UNUSABLE. This setting allows all operations (inserts, deletes, updates, and selects) on tables with unusable indexes or index partitions.

Note: If an index is used to enforce a UNIQUE constraint on a table, then allowing insert and update operations on the table might violate the constraint. Therefore, this setting does not disable error reporting for unusable indexes that are unique.

- false

Enables error reporting of indexes marked UNUSABLE. This setting does not allow inserts, deletes, and updates on tables with unusable indexes or index partitions.

SMTP_OUT_SERVER

Property	Description
Parameter type	String
Syntax	SMTP_OUT_SERVER = <i>server_clause</i> [, <i>server_clause</i>] ... server_clause::= <i>host_name</i> [: <i>port</i>]
Default value	There is no default value.
Modifiable	No

SMTP_OUT_SERVER specifies the SMTP host and port to which UTL_MAIL delivers out-bound E-mail. Multiple servers may be specified, separated by commas.

If the first server in the list is unavailable, then UTL_MAIL tries the second server, and so on.

If SMTP_OUT_SERVER is not specified, then the SMTP server name defaults to the value of DB_DOMAIN, the port number defaults to 25, and the SMTP domain defaults to the suffix of DB_DOMAIN.

See Also: *PL/SQL Packages and Types Reference* for information on the UTL_MAIL package

SORT_AREA_RETAINED_SIZE

Property	Description
Parameter type	Integer
Default value	Derived from SORT_AREA_SIZE
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	From the value equivalent of two database blocks to the value of SORT_AREA_SIZE

Note: Oracle does not recommend using the SORT_AREA_RETAINED_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. SORT_AREA_RETAINED_SIZE is retained for backward compatibility.

SORT_AREA_RETAINED_SIZE specifies (in bytes) the maximum amount of the user global area (UGA) memory retained after a sort run completes. The retained size controls the size of the read buffer, which Oracle uses to maintain a portion of the sort in memory. This memory is released back to the UGA, not to the operating system, after the last row is fetched from the sort space.

Oracle may allocate multiple sort spaces of this size for each query. Usually, only one or two sorts occur at one time, even for complex queries. In some cases, however, additional concurrent sorts are required, and each sort keeps its own memory area. If the shared server is used, allocation is to the SGA until the value in `SORT_AREA_RETAINED_SIZE` is reached. The difference between `SORT_AREA_RETAINED_SIZE` and `SORT_AREA_SIZE` is allocated to the PGA.

Note: The default value as reflected in the `V$PARAMETER` dynamic performance view is 0. However, if you do not explicitly set this parameter, Oracle actually uses the value of the `SORT_AREA_SIZE` parameter.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "[SORT_AREA_SIZE](#)" on page 1-131

SORT_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	65536
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	Minimum: the value equivalent of six database blocks Maximum: operating system-dependent

Note: Oracle does not recommend using the `SORT_AREA_SIZE` parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting `PGA_AGGREGATE_TARGET` instead. `SORT_AREA_SIZE` is retained for backward compatibility.

`SORT_AREA_SIZE` specifies (in bytes) the maximum amount of memory Oracle will use for a sort. After the sort is complete, but before the rows are returned, Oracle releases all of the memory allocated for the sort, except the amount specified by the `SORT_AREA_RETAINED_SIZE` parameter. After the last row is returned, Oracle releases the remainder of the memory.

Increasing `SORT_AREA_SIZE` size improves the efficiency of large sorts. Multiple allocations never occur. Only one memory area of `SORT_AREA_SIZE` exists for each user process at any time.

`SORT_AREA_SIZE` is also used for inserts and updates to bitmap indexes. Setting this value appropriately results in a bitmap segment being updated only once for each DML operation, even if more than one row in that segment changes.

Larger values of `SORT_AREA_SIZE` permit more sorts to be performed in memory. If more space is required to complete the sort than will fit into the memory provided, then temporary segments on disk are used to hold the intermediate sort runs.

The default is adequate for most OLTP operations. You might want to adjust this parameter for decision support systems, batch jobs, or large `CREATE INDEX` operations.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Concepts* for information on logical storage structures such as sort areas
- Your operating system-specific Oracle documentation for the default value on your system
- ["SORT_AREA_RETAINED_SIZE"](#) on page 1-130

SPFILE

Property	Description
Parameter type	String
Syntax	<code>SPFILE = spfile_name</code>
Default value	<code>ORACLE_HOME/dbs/spfile.ora</code>
Modifiable	No
Range of values	Any valid <code>SPFILE</code>
Real Application Clusters	Multiple instances should have the same value.

The value of this parameter is the name of the current server parameter file (`SPFILE`) in use. This parameter can be defined in a client side `PFILE` to indicate the name of the server parameter file to use.

When the default server parameter file is used by the server, the value of `SPFILE` is internally set by the server.

The `SPFILE` resides in the `ORACLE_HOME/dbs` directory; however, users can place it anywhere on their machine as long as it is specified in an initialization parameter file.

See Also: *Oracle Database Administrator's Guide* for more information about creating the server parameter file

SQL_TRACE

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	<code>ALTER SESSION, ALTER SYSTEM</code>
Range of values	true false

SQL_TRACE enables or disables the SQL trace facility. Setting this parameter to `true` provides information on tuning that you can use to improve performance. You can change the value using the `DBMS_SYSTEM` package.

Caution: Using this initialization parameter to enable the SQL trace facility for the entire instance can have a severe performance impact. Enable the facility for specific sessions using the `ALTER SESSION` statement. If you must enable the facility on an entire production environment, then you can minimize performance impact by:

- Maintaining at least 25% idle CPU capacity
 - Maintaining adequate disk space for the `USER_DUMP_DEST` location
 - Striping disk space over sufficient disks
-

See Also: *Oracle Database Performance Tuning Guide* for more information about performance diagnostic tools

SQL92_SECURITY

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true false

The SQL92 standards specify that security administrators should be able to require that users have `SELECT` privilege on a table when executing an `UPDATE` or `DELETE` statement that references table column values in a `WHERE` or `SET` clause. `SQL92_SECURITY` specifies whether users must have been granted the `SELECT` object privilege in order to execute such `UPDATE` or `DELETE` statements.

SQLTUNE_CATEGORY

Property	Description
Parameter type	String
Syntax	<code>SQLTUNE_CATEGORY = category_name</code>
Default value	DEFAULT
Modifiable	<code>ALTER SESSION</code> , <code>ALTER SYSTEM</code>

`SQLTUNE_CATEGORY` specifies the category name for use by sessions to qualify the lookup of SQL profiles during SQL compilation.

See Also: *PL/SQL Packages and Types Reference* for information on the `DBMS_SQLTUNE` package

STANDBY_ARCHIVE_DEST

Property	Description
Parameter type	String
Syntax	STANDBY_ARCHIVE_DEST = <i>filespec</i>
Default value	Operating system-specific
Modifiable	ALTER SYSTEM
Range of values	A valid path or device name other than RAW

STANDBY_ARCHIVE_DEST is relevant only for a standby database in managed recovery mode. It specifies the location of archive logs arriving from a primary database. Oracle uses STANDBY_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT to fabricate the fully qualified standby log filenames and stores the filenames in the standby control file.

You can see the value of this parameter by querying the V\$ARCHIVE_DEST data dictionary view.

See Also:

- *Oracle Data Guard Concepts and Administration* for more information on setting this parameter and on managed recovery mode in general
- "[LOG_ARCHIVE_DEST](#)" on page 1-70 and "[V\\$ARCHIVE_DEST](#)" on page 4-6

STANDBY_FILE_MANAGEMENT

Property	Description
Parameter type	String
Syntax	STANDBY_FILE_MANAGEMENT = { MANUAL AUTO }
Default value	MANUAL
Modifiable	ALTER SYSTEM

STANDBY_FILE_MANAGEMENT enables or disables automatic standby file management. When automatic standby file management is enabled, operating system file additions and deletions on the primary database are replicated on the standby database.

Values:

- MANUAL
 - disables automatic standby file management
- AUTO
 - enables automatic standby file management

Setting STANDBY_FILE_MANAGEMENT to AUTO causes Oracle to automatically create files on the standby database and, in some cases, overwrite existing files. Care must be

taken when setting `STANDBY_FILE_MANAGEMENT` and `DB_FILE_NAME_CONVERT` so that existing standby files will not be accidentally overwritten.

If the standby database is on the same system as the primary database, then ensure that the primary and standby systems do not point to the same files.

See Also: *Oracle Data Guard Concepts and Administration* for more information about setting this parameter

STAR_TRANSFORMATION_ENABLED

Property	Description
Parameter type	String
Syntax	<code>STAR_TRANSFORMATION_ENABLED = {TEMP_DISABLE TRUE FALSE}</code>
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	Yes

`STAR_TRANSFORMATION_ENABLED` determines whether a cost-based query transformation will be applied to star queries.

Values:

- TRUE
The optimizer will consider performing a cost-based query transformation on the star query.
- FALSE
The transformation will not be applied.
- TEMP_DISABLE
The optimizer will consider performing a cost-based query transformation on the star query but will not use temporary tables in the star transformation.

See Also:

- *Oracle Database Concepts* for information on star queries
- *Oracle Database Performance Tuning Guide* for information on enabling star query

STATISTICS_LEVEL

Property	Description
Parameter type	String
Syntax	<code>STATISTICS_LEVEL = { ALL TYPICAL BASIC }</code>
Default value	TYPICAL
Modifiable	ALTER SESSION, ALTER SYSTEM

STATISTICS_LEVEL specifies the level of collection for database and operating system statistics. The Oracle Database collects these statistics for a variety of purposes, including making self-management decisions.

The default setting of TYPICAL ensures collection of all major statistics required for database self-management functionality and provides best overall performance. The default value should be adequate for most environments.

When the STATISTICS_LEVEL parameter is set to ALL, additional statistics are added to the set of statistics collected with the TYPICAL setting. The additional statistics are timed OS statistics and plan execution statistics.

Setting the STATISTICS_LEVEL parameter to BASIC disables the collection of many of the important statistics required by Oracle Database features and functionality, including:

- Automatic Workload Repository (AWR) Snapshots
- Automatic Database Diagnostic Monitor (ADDM)
- All server-generated alerts
- Automatic SGA Memory Management
- Automatic optimizer statistics collection
- Object level statistics
- End to End Application Tracing (V\$CLIENT_STATS)
- Database time distribution statistics (V\$SESS_TIME_MODEL and V\$SYS_TIME_MODEL)
- Service level statistics
- Buffer cache advisory
- MTTR advisory
- Shared pool sizing advisory
- Segment level statistics
- PGA Target advisory
- Timed statistics
- Monitoring of statistics

Note: Oracle strongly recommends that you do not disable these important features and functionality.

When the STATISTICS_LEVEL parameter is modified by ALTER SYSTEM, all advisories or statistics are dynamically turned on or off, depending on the new value of STATISTICS_LEVEL. When modified by ALTER SESSION, the following advisories or statistics are turned on or off in the local session only. Their systemwide state is not changed:

- Timed statistics
- Timed OS statistics
- Plan execution statistics

The `V$STATISTICS_LEVEL` view displays information about the status of the statistics or advisories controlled by the `STATISTICS_LEVEL` parameter. See "[V\\$STATISTICS_LEVEL](#)" on page 5-71.

STREAMS_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>STREAMS_POOL_SIZE = integer [K M G]</code>
Default value	0
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent

`STREAMS_POOL_SIZE` specifies (in bytes) the size of the Streams pool, from which memory is allocated for Streams. If this parameter is not specified or is set to 0, then up to 10% of the shared pool is allocated for Streams.

See Also: *Oracle Streams Concepts and Administration* for more information on setting this parameter

TAPE_ASYNC_IO

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Range of values	true false

`TAPE_ASYNC_IO` controls whether I/O to sequential devices (for example, backup or restore of Oracle data to or from tape) is asynchronous—that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans. If your platform supports asynchronous I/O to sequential devices, Oracle recommends that you leave this parameter set to its default. However, if the asynchronous I/O implementation is not stable, you can set `TAPE_ASYNC_IO` to `false` to disable asynchronous I/O. If your platform does not support asynchronous I/O to sequential devices, this parameter has no effect.

See Also: *Oracle Database Performance Tuning Guide* for more information on asynchronous I/O and on setting this parameter

THREAD

Property	Description
Parameter type	Integer

Property	Description
Default value	0
Modifiable	No
Range of values	0 to the maximum number of enabled threads
Real Application Clusters	If specified, multiple instances must have different values.

THREAD is a Real Application Clusters parameter that specifies the number of the redo thread to be used by an instance.

When you create a database, Oracle creates and enables thread 1 as a **public thread** (one that can be used by any instance). You must create and enable subsequent threads using the `ADD LOGFILE THREAD` clause and `ENABLE THREAD` clause of the `ALTER DATABASE` statement. The number of threads you create is limited by the `MAXINSTANCES` parameter specified in the `CREATE DATABASE` statement.

In exclusive mode, thread 1 is the default thread. However, you can specify `THREAD` for an instance running in exclusive mode if you want to use the redo log files in a thread other than thread 1.

In parallel mode, you can specify any available redo thread number, as long as that thread number is enabled and is not in use by another instance.

A value of zero specifies that this instance can use any available, enabled public thread.

See Also: *Oracle Real Application Clusters Administrator's Guide* and *Oracle Database SQL Reference*

TIMED_OS_STATISTICS

Property	Description
Parameter type	Integer
Default value	If <code>STATISTICS_LEVEL</code> is set to <code>ALL</code> , then 5 If <code>STATISTICS_LEVEL</code> is set to <code>BASIC</code> or <code>TYPICAL</code> , then 0
Modifiable	<code>ALTER SESSION</code> , <code>ALTER SYSTEM</code>
Range of values	Unlimited

`TIMED_OS_STATISTICS` specifies (in seconds) the interval at which Oracle collects operating system statistics when a request is made from the client to the server or when a request completes.

- On dedicated servers, Oracle collects operating system statistics at user logon and after each subsequent client invocation through the OCI into the Oracle server as a remote procedure call message.
- On shared servers, Oracle collects statistics when client calls to Oracle are processed.

A value of zero specifies that operating system statistics are not gathered. To collect statistics, set a value meaningful for your application and site needs.

Note: Gathering operating system statistics is very expensive. Oracle recommends that you set this parameter in an `ALTER SYSTEM` statement rather than in the initialization parameter file, and that you reset the value to zero as soon as the needed statistics have been gathered.

TIMED_STATISTICS

Property	Description
Parameter type	Boolean
Default value	If <code>STATISTICS_LEVEL</code> is set to <code>TYPICAL</code> or <code>ALL</code> , then <code>true</code> If <code>STATISTICS_LEVEL</code> is set to <code>BASIC</code> , then <code>false</code>
Modifiable	<code>ALTER SESSION</code> , <code>ALTER SYSTEM</code>
Range of values	<code>true</code> <code>false</code>

`TIMED_STATISTICS` specifies whether or not statistics related to time are collected.

Values:

- `true`
The statistics are collected and stored in trace files or displayed in the `V$SESSTATS` and `V$SYSSTATS` dynamic performance views.
- `false`
The value of all time-related statistics is set to zero. This setting lets Oracle avoid the overhead of requesting the time from the operating system. Normally, `TIMED_STATISTICS` should be `false`.

On some systems with very fast timer access, Oracle might enable timing even if this parameter is set to `false`. On these systems, setting the parameter to `true` can sometimes produce more accurate statistics for long-running operations.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter and on performance diagnostic tools in general
- [Appendix E, "Statistics Descriptions"](#) indicates which statistics depend on the setting of this parameter.

TRACE_ENABLED

Property	Description
Parameter type	Boolean
Default value	<code>true</code>
Modifiable	<code>ALTER SYSTEM</code>
Range of values	<code>true</code> <code>false</code>

Property	Description
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

TRACE_ENABLED controls tracing of the execution history, or code path, of Oracle. Oracle Support Services uses this information for debugging.

When TRACE_ENABLED is set to `true`, Oracle records information in specific files when errors occur. See *Oracle Real Application Clusters Deployment and Performance Guide* for the types of files and the default destination directories in which Oracle records the execution history.

Oracle records this information for all instances, even if only one instance terminates. This allows Oracle to retain diagnostics for the entire cluster.

Although the overhead incurred from this processing is not excessive, you can improve performance by setting TRACE_ENABLED to `false`. You might do this, for example, to meet high-end benchmark requirements. However, if you leave this parameter set to `false`, you may lose valuable diagnostic information. Therefore, always set TRACE_ENABLED to `true` to trace system problems and to reduce diagnostic efforts in the event of unexplained instance failures.

TRACEFILE_IDENTIFIER

Property	Description
Parameter type	String
Syntax	TRACEFILE_IDENTIFIER = " <i>traceid</i> "
Default value	There is no default value.
Modifiable	ALTER SESSION
Range of values	Any characters that can occur as part of a file name on the customer platform

TRACEFILE_IDENTIFIER specifies a custom identifier that becomes part of the Oracle Trace file name. Such a custom identifier is used to identify a trace file simply from its name and without having to open it or view its contents.

Each time this parameter is dynamically modified, the next trace dump will be written to a trace file which has the new parameter value embedded in its name. Trace file continuity information is automatically added to both the old and new trace files to indicate that these trace files belong to the same process.

This parameter can only be used to change the name of the foreground process' trace file; the background processes continue to have their trace files named in the regular format. For foreground processes, the TRACEID column of the V\$PROCESS view contains the current value of the TRACEFILE_IDENTIFIER parameter. When this parameter value is set, the trace file name has the following format:

```
sid_ora_pid_traceid.trc
```

In this example, *sid* is the oracle instance ID, *pid* is the process ID, and *traceid* is the value of the TRACEFILE_IDENTIFIER parameter.

See Also: This parameter is not supported on all operating systems. See your operating system-specific Oracle documentation for more information.

TRANSACTIONS

Property	Description
Parameter type	Integer
Default value	Derived: (1.1 * SESSIONS)
Modifiable	No
Range of values	4 to 2 ³²
Real Application Clusters	Multiple instances can have different values.

TRANSACTIONS specifies the maximum number of concurrent transactions. Greater values increase the size of the SGA and can increase the number of rollback segments allocated. The default value is greater than SESSIONS (and, in turn, PROCESSES) to allow for recursive transactions.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter
- *Oracle Real Application Clusters Administrator's Guide* for the relationship of this parameter to the number of rollback segments

TRANSACTIONS_PER_ROLLBACK_SEGMENT

Property	Description
Parameter type	Integer
Default value	5
Modifiable	No
Range of values	1 to operating system-dependent
Real Application Clusters	Multiple instances can have different values.

TRANSACTIONS_PER_ROLLBACK_SEGMENT specifies the number of concurrent transactions you expect each rollback segment to have to handle. The minimum number of rollback segments acquired at startup is TRANSACTIONS divided by the value for this parameter. For example, if TRANSACTIONS is 101 and this parameter is 10, then the minimum number of rollback segments acquired would be the ratio 101/10, rounded up to 11.

You can acquire more rollback segments by naming them in the parameter ROLLBACK_SEGMENTS.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter.
- *Oracle Real Application Clusters Administrator's Guide* for how Oracle acquires rollback segments.
- Your operating system-specific Oracle documentation for the range of values for this parameter.

UNDO_MANAGEMENT

Property	Description
Parameter type	String
Syntax	UNDO_MANAGEMENT = { MANUAL AUTO }
Default value	MANUAL
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

UNDO_MANAGEMENT specifies which undo space management mode the system should use. When set to AUTO, the instance starts in automatic undo management mode. In manual undo management mode, undo space is allocated externally as rollback segments.

UNDO_RETENTION

Property	Description
Parameter type	Integer
Default value	900
Modifiable	ALTER SYSTEM
Range of values	0 to $2^{32} - 1$ (max value represented by 32 bits)
Real Application Clusters	Multiple instances must have the same value.

UNDO_RETENTION specifies (in seconds) the low threshold value of undo retention. The system retains undo for at least the time specified in this parameter and automatically tunes the undo retention period to satisfy the undo requirements of the queries.

The setting of this parameter should account for any flashback requirements of the system. Automatic tuning of undo retention is not supported for LOBs. The RETENTION value for LOB columns is set to the value of the UNDO_RETENTION parameter.

The UNDO_RETENTION parameter can only be honored if the current undo tablespace has enough space. If an active transaction requires undo space and the undo tablespace does not have available space, then the system starts reusing unexpired

undo space. This action can potentially cause some queries to fail with a "snapshot too old" message.

The amount of time for which undo is retained for the Oracle Database for the current undo tablespace can be obtained by querying the `TUNED_UNDORETENTION` column of the `V$UNDOSTAT` dynamic performance view.

See Also:

- *Oracle Database SQL Reference* for more information about creating undo tablespaces
- *Oracle Database Administrator's Guide* for more information about managing undo

UNDO_TABLESPACE

Property	Description
Parameter type	String
Syntax	<code>UNDO_TABLESPACE = undoname</code>
Default value	The first available undo tablespace in the database.
Modifiable	<code>ALTER SYSTEM</code>
Range of values	Legal name of an existing undo tablespace
Basic	Yes
Real Application Clusters	Multiple instances can have different values.

`UNDO_TABLESPACE` specifies the undo tablespace to be used when an instance starts up. If this parameter is specified when the instance is in manual undo management mode, then an error will occur and startup will fail.

If the `UNDO_TABLESPACE` parameter is omitted, the first available undo tablespace in the database is chosen. If no undo tablespace is available, the instance will start without an undo tablespace. In such cases, user transactions will be executed using the `SYSTEM` rollback segment. You should avoid running in this mode under normal circumstances.

You can replace an undo tablespace with another undo tablespace while the instance is running.

See Also: *Oracle Database SQL Reference*.

USE_INDIRECT_DATA_BUFFERS

Property	Description
Parameter type	Boolean
Default value	<code>false</code>
Modifiable	No
Range of values	<code>true</code> <code>false</code>

USE_INDIRECT_DATA_BUFFERS controls how the system global area (SGA) uses memory. It enables or disables the use of the extended buffer cache mechanism for 32-bit platforms that can support more than 4 GB of physical memory. On platforms that do not support this much physical memory, this parameter is ignored.

See Also:

- "LOCK_SGA" on page 1-69, "SHARED_MEMORY_ADDRESS" on page 1-127, and "HI_SHARED_MEMORY_ADDRESS" on page 1-59, which are other parameters that control how the SGA uses memory.
- *Oracle Database Concepts* for more information about the SGA.

USER_DUMP_DEST

Property	Description
Parameter type	String
Syntax	USER_DUMP_DEST = { <i>pathname</i> <i>directory</i> }
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM
Range of values	Any valid local path, directory, or disk

USER_DUMP_DEST specifies the pathname for a directory where the server will write debugging trace files on behalf of a user process.

For example, this directory might be set as follows:

- On MS-DOS: C:\ORACLE\UTRC
- On UNIX: /oracle/utrc
- On VMS: DISK\$UR3:[ORACLE.UTRC]

See Also:

- *Oracle Database Performance Tuning Guide* for more information about the use of trace files
- Your operating system-specific Oracle documentation for the range of values

UTL_FILE_DIR

Property	Description
Parameter type	String
Syntax	UTL_FILE_DIR = <i>pathname</i>
Default value	There is no default value.
Modifiable	No
Range of values	Any valid directory path

UTL_FILE_DIR lets you specify one or more directories that Oracle should use for PL/SQL file I/O. If you are specifying multiple directories, you must repeat the UTL_FILE_DIR parameter for each directory on separate lines of the initialization parameter file.

All users can read or write to all files specified by this parameter. Therefore all PL/SQL users must be trusted with the information in the directories specified by this parameter.

Note: If you list multiple values, all entries of this parameter must be on contiguous lines of the parameter file. If you separate them with other parameters, Oracle will read only the last (contiguous) lines.

WORKAREA_SIZE_POLICY

Property	Description
Parameter type	String
Syntax	WORKAREA_SIZE_POLICY = { AUTO MANUAL }
Default value	AUTO
Modifiable	ALTER SESSION, ALTER SYSTEM

WORKAREA_SIZE_POLICY specifies the policy for sizing work areas. This parameter controls the mode in which working areas are tuned.

Values:

- AUTO

Work areas used by memory-intensive operators are sized automatically, based on the PGA memory used by the system, the target PGA memory set in PGA_AGGREGATE_TARGET, and the requirement of each individual operator. You can specify AUTO only when PGA_AGGREGATE_TARGET is defined.

- MANUAL

The sizing of work areas is manual and based on the values of the *_AREA_SIZE parameter corresponding to the operation (for example, a sort uses SORT_AREA_SIZE). Specifying MANUAL may result in sub-optimal performance and poor PGA memory utilization.

Part II

Static Data Dictionary Views

This part describes data dictionary tables and views. These tables and views are called **static**, because they change only when a change is made to the data dictionary (for example, when a new table is created or when a user is granted new privileges). This part contains the following chapters:

- [Chapter 2, "Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES"](#)
- [Chapter 3, "Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS"](#)

Note: Oracle also maintains tables that monitor ongoing database activity. These **dynamic performance tables** are described in [Part III, "Dynamic Performance Views"](#).

Static Data Dictionary Views: ALL_ALL_TABLES to DATABASE_PROPERTIES

This chapter describes the first set (in alphabetical order) of static data dictionary views. The remaining static data dictionary views appear in alphabetical order in [Chapter 3](#).

This chapter contains the following topics:

- [About Static Data Dictionary Views](#)
- [Static Data Dictionary View Descriptions](#)

About Static Data Dictionary Views

Data dictionary tables are not directly accessible, but you can access information in them through data dictionary views. To list the data dictionary views available to you, query the view `DICTIONARY`.

Many data dictionary tables have three corresponding views:

- An `ALL_` view displays all the information accessible to the current user, including information from the current user's schema as well as information from objects in other schemas, if the current user has access to those objects by way of grants of privileges or roles.
- A `DBA_` view displays all relevant information in the entire database. `DBA_` views are intended only for administrators. They can be accessed only by users with the `SELECT ANY TABLE` privilege. This privilege is assigned to the `DBA` role when the system is initially installed.
- A `USER_` view displays all the information from the schema of the current user. No special privileges are required to query these views.

The columns of the ALL_, DBA_, and USER_ views corresponding to a single data dictionary table are usually nearly identical. Therefore, these views are described in full only once in this chapter, at their first occurrence alphabetically, and are listed without full descriptions at their other occurrences.

Oracle Replication Views

A number of data dictionary views are relevant only if you are using Oracle Replication. [Table 2–1](#) lists these views.

Table 2–1 Oracle Replication Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_REPCAT	DBA_REPCAT	USER_REPCAT
	DBA_REPCAT_EXCEPTIONS	
ALL_REPCAT_REFRESH_TEMPLATES	DBA_REPCAT_REFRESH_TEMPLATES	USER_REPCAT_REFRESH_TEMPLATES
ALL_REPCAT_TEMPLATE_OBJECTS	DBA_REPCAT_TEMPLATE_OBJECTS	USER_REPCAT_TEMPLATE_OBJECTS
ALL_REPCAT_TEMPLATE_PARS	DBA_REPCAT_TEMPLATE_PARS	USER_REPCAT_TEMPLATE_PARS
ALL_REPCAT_TEMPLATE_SITES	DBA_REPCAT_TEMPLATE_SITES	USER_REPCAT_TEMPLATE_SITES
ALL_REPCAT_USER_AUTHORIZATIONS	DBA_REPCAT_USER_AUTHORIZATIONS	USER_REPCAT_USER_AUTHORIZATIONS
ALL_REPCAT_USER_PARM_VALUES	DBA_REPCAT_USER_PARM_VALUES	USER_REPCAT_USER_PARM_VALUES
ALL_REPCATLOG	DBA_REPCATLOG	USER_REPCATLOG
ALL_REPCOLUMN	DBA_REPCOLUMN	USER_REPCOLUMN
ALL_REPCOLUMN_GROUP	DBA_REPCOLUMN_GROUP	USER_REPCOLUMN_GROUP
ALL_REPCONFLICT	DBA_REPCONFLICT	USER_REPCONFLICT
ALL_REPDDL	DBA_REPDDL	USER_REPDDL
	DBA_REPEXTENSIONS	
ALL_REPFLAVOR_COLUMNS	DBA_REPFLAVOR_COLUMNS	USER_REPFLAVOR_COLUMNS
ALL_REPFLAVOR_OBJECTS	DBA_REPFLAVOR_OBJECTS	USER_REPFLAVOR_OBJECTS
ALL_REPFLAVORS	DBA_REPFLAVORS	USER_REPFLAVORS
ALL_REPGENERATED	DBA_REPGENERATED	USER_REPGENERATED
ALL_REPGENOBJECTS	DBA_REPGENOBJECTS	USER_REPGENOBJECTS
ALL_REPGROUP	DBA_REPGROUP	USER_REPGROUP
ALL_REPGROUP_PRIVILEGES	DBA_REPGROUP_PRIVILEGES	USER_REPGROUP_PRIVILEGES
ALL_REPGROUPED_COLUMN	DBA_REPGROUPED_COLUMN	USER_REPGROUPED_COLUMN
ALL_REPKEY_COLUMNS	DBA_REPKEY_COLUMNS	USER_REPKEY_COLUMNS
ALL_REPOBJECT	DBA_REPOBJECT	USER_REPOBJECT
ALL_REPPARAMETER_COLUMN	DBA_REPPARAMETER_COLUMN	USER_REPPARAMETER_COLUMN
ALL_REPPRIORITY	DBA_REPPRIORITY	USER_REPPRIORITY
ALL_REPPRIORITY_GROUP	DBA_REPPRIORITY_GROUP	USER_REPPRIORITY_GROUP
ALL_REPPROP	DBA_REPPROP	USER_REPPROP
ALL_REPRESOL_STATS_CONTROL	DBA_REPRESOL_STATS_CONTROL	USER_REPRESOL_STATS_CONTROL
ALL_REPRESOLUTION	DBA_REPRESOLUTION	USER_REPRESOLUTION
ALL_REPRESOLUTION_METHOD	DBA_REPRESOLUTION_METHOD	USER_REPRESOLUTION_METHOD
ALL_REPRESOLUTION_STATISTICS	DBA_REPRESOLUTION_STATISTICS	USER_REPRESOLUTION_STATISTICS

Table 2–1 (Cont.) Oracle Replication Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_REPSHEMA	DBA_REPSHEMA	USER_REPSHEMA
ALL_REPSITES	DBA_REPSITES	USER_REPSITES
	DBA_REPSITES_NEW	

The following are additional Oracle Replication data dictionary views:

DEFCALL
 DEFCALLDEST
 DEFDEFAULTDEST
 DEFERRCOUNT
 DEFERROR
 DEFLOB
 DEFPROPAGATOR
 DEFSCHEDULE
 DEFTRAN
 DEFTRANDEST

See Also: *Oracle Database Advanced Replication Management API Reference* for information about these views

Oracle Workspace Manager Views

A number of data dictionary views are relevant only if you are using Oracle Workspace Manager. [Table 2–2](#) lists these views.

Table 2–2 Oracle Workspace Manager Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_VERSION_HVIEW		
ALL_WM_LOCKED_TABLES		USER_WM_LOCKED_TABLES
ALL_WM_MODIFIED_TABLES		USER_WM_MODIFIED_TABLES
		USER_WM_PRIVS
ALL_WM_RIC_INFO		USER_WM_RIC_INFO
ALL_WM_TAB_TRIGGERS		USER_WM_TAB_TRIGGERS
ALL_WM_VERSIONED_TABLES		USER_WM_VERSIONED_TABLES
ALL_WM_VT_ERRORS		USER_WM_VT_ERRORS
ALL_WORKSPACE_PRIVS		USER_WORKSPACE_PRIVS
ALL_WORKSPACE_SAVEPOINTS		USER_WORKSPACE_SAVEPOINTS
	DBA_WORKSPACE_SESSIONS	
ALL_WORKSPACES		USER_WORKSPACES

The following are additional Oracle Workspace Manager data dictionary views:

ROLE_WM_PRIVS
 WM_INSTALLATION
 WM_REPLICATION_INFO

See Also: *Oracle Database Application Developer's Guide - Workspace Manager* for information about these views

Recovery Catalog Views

The following data dictionary views are only available after you create an optional recovery catalog (which contains schemas containing information about backups) for use with Recovery Manager:

RC_ARCHIVED_LOG
RC_BACKUP_CONTROLFILE
RC_BACKUP_CORRUPTION
RC_BACKUP_DATAFILE
RC_BACKUP_PIECE
RC_BACKUP_REDOLOG
RC_BACKUP_SET
RC_BACKUP_SPFILE
RC_CHECKPOINT
RC_CONTROLFILE_COPY
RC_COPY_CORRUPTION
RC_DATABASE
RC_DATABASE_BLOCK_CORRUPTION
RC_DATABASE_INCARNATION
RC_DATAFILE
RC_DATAFILE_COPY
RC_LOG_HISTORY
RC_OFFLINE_RANGE
RC_PROXY_CONTROLFILE
RC_PROXY_DATAFILE
RC_REDO_LOG
RC_REDO_THREAD
RC_RESYNC
RC_RMAN_CONFIGURATION
RC_STORED_SCRIPT
RC_STORED_SCRIPT_LINE
RC_TABLESPACE

See Also: *Oracle Database Recovery Manager Reference* for information about these views

Static Data Dictionary View Descriptions

The remainder of this chapter describes the static data dictionary views in alphabetical order.

ALL_ALL_TABLES

ALL_ALL_TABLES describes the object tables and relational tables accessible to the current user.

Related Views

- DBA_ALL_TABLES describes all object tables and relational tables in the database.

- USER_ALL_TABLES describes the object tables and relational tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; null for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
PCT_FREE	NUMBER		Minimum percentage of free space in a block; null for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; null for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; null for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; null for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); null for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); null for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; null for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; null for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; null for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; null for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Indicates whether the table has been backed up since the last modification (Y) or not (N)
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of rows in the table that are chained from one data block to another or that have migrated to a new block, requiring a link to preserve the old rowid. This column is updated only after you analyze the table.
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)

Column	Datatype	NULL	Description
TABLE_LOCK	VARCHAR2(8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the table
LAST_ANALYZED	DATE		Date on which the table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Indicates whether the table is partitioned (YES) or not (NO)
IOT_TYPE	VARCHAR2(12)		If the table is an index-organized table, indicates whether an overflow segment has been specified (IOT_OVERFLOW) or not (NULL)
OBJECT_ID_TYPE	VARCHAR2(16)		Indicates whether the object ID is USER-DEFINED or SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		If an object table, owner of the type from which the table is created
TABLE_TYPE	VARCHAR2(30)		If an object table, type of the table
TEMPORARY	VARCHAR2(1)		Indicates whether the table is temporary (Y) or not (N)
SECONDARY	VARCHAR2(1)		Indicates whether the table is a secondary object created by the ODCI IndexCreate method of the Oracle Data Cartridge to contain the contents of a domain index (Y) or not (N)
NESTED	VARCHAR2(3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2(7)		Buffer pool to be used for table blocks: <ul style="list-style-type: none"> ■ DEFAULT ■ KEEP ■ RECYCLE
ROW_MOVEMENT	VARCHAR2(8)		If a partitioned table, indicates whether row movement is enabled (ENABLED) or disabled (DISABLED)
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected by analyzing the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: SYS\$SESSION - Rows are preserved for the duration of the session SYS\$TRANSACTION - Rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2(8)		Indicates whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.skip_corrupt_blocks procedure.
MONITORING	VARCHAR2(3)		Indicates whether the table has the MONITORING attribute set (YES) or not (NO)
CLUSTER_OWNER	VARCHAR2(30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2(8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); null for partitioned tables
DROPPED	VARCHAR2(3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); null for partitioned tables

See Also:

- ["DBA_ALL_TABLES"](#) on page 3-18
- ["USER_ALL_TABLES"](#) on page 3-159

ALL_APPLY

ALL_APPLY displays information about the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY displays information about all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue from which the apply process dequeues
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue from which the apply process dequeues
APPLY_CAPTURED	VARCHAR2(3)		Indicates whether the apply process applies captured events (YES) or user-enqueued events (NO)
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set used by the apply process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set used by the apply process for filtering
APPLY_USER	VARCHAR2(30)		User who is applying events
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.
APPLY_TAG	RAW(2000)		Tag associated with redo log records that are generated when changes are made by the apply process
DDL_HANDLER	VARCHAR2(98)		Name of the user-specified DDL handler, which handles DDL logical change records
PRECOMMIT_HANDLER	VARCHAR2(98)		Name of the user-specified pre-commit handler
MESSAGE_HANDLER	VARCHAR2(98)		Name of the user-specified procedure that handles dequeued events other than logical change records
STATUS	VARCHAR2(8)		Status of the apply process: <ul style="list-style-type: none"> ■ DISABLED ■ ENABLED ■ ABORTED
MAX_APPLIED_MESSAGE_NUMBER	NUMBER		Maximum value of the message that has been applied
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set used by the apply process for filtering
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set used by the apply process for filtering
STATUS_CHANGE_TIME	DATE		Time that the STATUS of the apply process was changed
ERROR_NUMBER	NUMBER		Error number if the apply process was aborted
ERROR_MESSAGE	VARCHAR2(4000)		Error message if the apply process was aborted

See Also: ["DBA_APPLY"](#) on page 3-19

ALL_APPLY_CONFLICT_COLUMNS

ALL_APPLY_CONFLICT_COLUMNS displays information about conflict handlers on the tables accessible to the current user.

Related View

DBA_APPLY_CONFLICT_COLUMNS displays information about conflict handlers on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the object on which the update conflict handler is defined
OBJECT_NAME	VARCHAR2(30)		Name of the object on which the update conflict handler is defined
METHOD_NAME	VARCHAR2(92)		Name of the update conflict handler used to resolve conflicts
RESOLUTION_COLUMN	VARCHAR2(4000)		Name of the column used to resolve conflicts
COLUMN_NAME	VARCHAR2(30)		Name of a column in the column list for the update conflict handler
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

See Also: ["DBA_APPLY_CONFLICT_COLUMNS"](#) on page 3-19

ALL_APPLY_DML_HANDLERS

ALL_APPLY_DML_HANDLERS displays information about the DML handlers on the tables accessible to the current user.

Related View

DBA_APPLY_DML_HANDLERS displays information about the DML handlers on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object on which the DML handler is specified
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object on which the DML handler is specified
OPERATION_NAME	VARCHAR2(13)		Name of the DML operation for which the DML handler is used
USER_PROCEDURE	VARCHAR2(98)		Name of the user-specified DML handler, which handles row logical change records that contain the DML operation in the OPERATION_NAME column on the object
ERROR_HANDLER	VARCHAR2(1)		Indicates whether the DML handler handles only the relevant row logical change records that result in apply errors (Y) or all relevant row logical change records (N)
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.
APPLY_NAME	VARCHAR2(30)		Name of the apply process for the given object

See Also: ["DBA_APPLY_DML_HANDLERS"](#) on page 3-19

ALL_APPLY_ENQUEUE

ALL_APPLY_ENQUEUE displays information about the apply enqueue actions for the rules where the destination queue exists and is accessible to the current user.

Related View

DBA_APPLY_ENQUEUE displays information about the apply enqueue actions for all rules in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
DESTINATION_QUEUE_NAME	VARCHAR2(4000)		Name of the queue where events satisfying the rule will be enqueued

See Also: ["DBA_APPLY_ENQUEUE"](#) on page 3-19

ALL_APPLY_ERROR

ALL_APPLY_ERROR displays information about error transactions generated by the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY_ERROR displays information about error transactions generated by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)		Name of the apply process at the local database which processed the transaction
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue at the local database from which the transaction was dequeued
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue at the local database from which the transaction was dequeued
LOCAL_TRANSACTION_ID	VARCHAR2(22)		Local transaction ID for the error transaction
SOURCE_DATABASE	VARCHAR2(128)		Database where the transaction originated
SOURCE_TRANSACTION_ID	VARCHAR2(22)		Original transaction ID at the source database
SOURCE_COMMIT_SCN	NUMBER		Original commit system change number (SCN) for the transaction at the source database
MESSAGE_NUMBER	NUMBER		Identifier for the event in the transaction that raised an error
ERROR_NUMBER	NUMBER		Error number of the error raised by the transaction
ERROR_MESSAGE	VARCHAR2(4000)		Error message of the error raised by the transaction
RECIPIENT_ID	NUMBER		User ID of the original user that applied the transaction
RECIPIENT_NAME	VARCHAR2(30)		Name of the original user that applied the transaction
MESSAGE_COUNT	NUMBER		Total number of events inside the error transaction

See Also: ["DBA_APPLY_ERROR"](#) on page 3-19

ALL_APPLY_EXECUTE

ALL_APPLY_EXECUTE displays information about the apply execute actions for the rules visible to the current user.

Related View

DBA_APPLY_EXECUTE displays information about the apply execute actions for all rules in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
EXECUTE_EVENT	VARCHAR2(2)		Whether the event satisfying the rule is executed

See Also: ["DBA_APPLY_EXECUTE"](#) on page 3-19

ALL_APPLY_KEY_COLUMNS

ALL_APPLY_KEY_COLUMNS displays information about substitute key columns for the tables accessible to the current user. Substitute key columns are set using the SET_KEY_COLUMNS procedure in the DBMS_APPLY_ADM package.

Related View

DBA_APPLY_KEY_COLUMNS displays information about substitute key columns for all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object on which substitute key columns are set
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object on which substitute key columns are set
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name of a column specified as a substitute key column
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

See Also: ["DBA_APPLY_KEY_COLUMNS"](#) on page 3-20

ALL_APPLY_PARAMETERS

ALL_APPLY_PARAMETERS displays information about the parameters for the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY_PARAMETERS displays information about the parameters for all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
PARAMETER	VARCHAR2(128)	NOT NULL	Name of the parameter

Column	Datatype	NULL	Description
VALUE	VARCHAR2(4000)		Parameter value
SET_BY_USER	VARCHAR2(3)		Indicates whether the parameter value was set by the user (YES) or was not set by the user (NO). If NO for a parameter, then the parameter is set to its default value. If YES for a parameter, then the parameter may or may not be set to its default value.

See Also: ["DBA_APPLY_PARAMETERS"](#) on page 3-21

ALL_APPLY_PROGRESS

ALL_APPLY_PROGRESS displays information about the progress made by the apply processes that dequeue events from queues accessible to the current user. This view only contains information about captured events. It does not contain information about user-enqueued events.

Related View

DBA_APPLY_PROGRESS displays information about the progress made by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Global name of the source database of the changes that are applied by the apply process
APPLIED_MESSAGE_NUMBER	NUMBER	NOT NULL	Message number up to which all transactions have definitely been applied. This value is the low-watermark for the apply process. That is, messages with a commit message number less than or equal to this message number have definitely been applied, but some messages with a higher commit message number also may have been applied.
OLDEST_MESSAGE_NUMBER	NUMBER	NOT NULL	Earliest message number of the transactions currently being dequeued and applied
APPLY_TIME	DATE		Time at which the message with the message number displayed in the APPLIED_MESSAGE_NUMBER column was applied
APPLIED_MESSAGE_CREATE_TIME	DATE		Time at which the message with the message number displayed in the APPLIED_MESSAGE_NUMBER column was created at its source database

See Also: ["DBA_APPLY_PROGRESS"](#) on page 3-21

ALL_APPLY_TABLE_COLUMNS

ALL_APPLY_TABLE_COLUMNS displays information about the destination table object columns for the tables accessible to the current user.

Related View

DBA_APPLY_TABLE_COLUMNS displays information about the destination table object columns for all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the column
COMPARE_OLD_ON_DELETE	VARCHAR2(3)		Indicates whether to Compare the old value of the column on deletes (YES) or not (NO)
COMPARE_OLD_ON_UPDATE	VARCHAR2(3)		Indicates whether to Compare the old value of the column on updates (YES) or not (NO)
APPLY_DATABASE_LINK	VARCHAR2(128)		For remote tables, name of the database link pointing to the remote database

See Also: ["DBA_APPLY_TABLE_COLUMNS"](#) on page 3-21

ALL_ARGUMENTS

ALL_ARGUMENTS lists the arguments of the procedures and functions that are accessible to the current user.

Related View

USER_ARGUMENTS lists the arguments of the procedures and functions that are owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the procedure or function
PACKAGE_NAME	VARCHAR2(30)		Name of the procedure or function
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
OVERLOAD	VARCHAR2(40)		Overload unique identifier
ARGUMENT_NAME	VARCHAR2(30)		Name of the argument
POSITION	NUMBER	NOT NULL	Position in the argument list, or null for a function return value
SEQUENCE	NUMBER	NOT NULL	Argument sequence, including all nesting levels
DATA_LEVEL	NUMBER	NOT NULL	Nesting depth of the argument for composite types
DATA_TYPE	VARCHAR2(30)		Datatype of the argument
DEFAULT_VALUE	LONG		Reserved for future use
DEFAULT_LENGTH	NUMBER		Reserved for future use
IN_OUT	VARCHAR2(9)		Direction of the argument: <ul style="list-style-type: none"> ▪ IN ▪ OUT ▪ IN/OUT
DATA_LENGTH	NUMBER		Length of the column (in bytes)
DATA_PRECISION	NUMBER		Length in decimal digits (NUMBER) or binary digits (FLOAT)
DATA_SCALE	NUMBER		Digits to the right of the decimal point in a number
RADIX	NUMBER		Argument radix for a number
CHARACTER_SET_NAME	VARCHAR2(44)		Character set name for the argument
TYPE_OWNER	VARCHAR2(30)		Owner of the type of the argument

Column	Datatype	NULL	Description
TYPE_NAME	VARCHAR2(30)		Name of the type of the argument. If the type is a package local type (that is, it is declared in a package specification), then this column displays the name of the package.
TYPE_SUBNAME	VARCHAR2(30)		Relevant only for package local types. Displays the name of the type declared in the package identified in the TYPE_NAME column.
TYPE_LINK	VARCHAR2(128)		Relevant only for package local types when the package identified in the TYPE_NAME column is a remote package. This column displays the database link used to refer to the remote package.
PLS_TYPE	VARCHAR2(30)		For numeric arguments, the name of the PL/SQL type of the argument. Null otherwise.
CHAR_LENGTH	NUMBER		Character limit for string datatypes
CHAR_USED	VARCHAR2(1)		Indicates whether the byte limit (B) or char limit (C) is official for the string

See Also: ["USER_ARGUMENTS"](#) on page 3-160

ALL_ASSOCIATIONS

ALL_ASSOCIATIONS describes user-defined statistics associated with objects accessible to the current user.

Related Views

- DBA_ASSOCIATIONS describes all user-defined statistics in the database.
- USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object for which the association is being defined
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object for which the association is being defined
COLUMN_NAME	VARCHAR2(30)		Column name in the object for which the association is being defined
OBJECT_TYPE	VARCHAR2(9)		Kind of object with which statistics are being associated: column, type, package or function, indextype, or domain index.
STATSTYPE_SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of statistics type that contains the cost, selectivity or statistics functions
DEF_SELECTIVITY	NUMBER		Default selectivity of the object, if any
DEF_CPU_COST	NUMBER		Default CPU cost of the object, if any
DEF_IO_COST	NUMBER		Default I/O cost of the object, if any
DEF_NET_COST	NUMBER		Default networking cost of the object, if any
INTERFACE_VERSION	NUMBER		Identifies the version number of the ODCIStats interface. Value is 1 for statistics type implementing Oracle8i 8.1; 0 for types implementing Oracle9i 9.0.0.

ALL_AUDIT_POLICIES

ALL_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views accessible to the current user.

Related Views

- DBA_AUDIT_POLICIES describes all fine-grained auditing policies in the database.
- USER_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views owned by the current user. This view does not display the OBJECT_SCHEMA column.

Column	Datatype	NULL	Description
OBJECT_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the table or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the table or view
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
POLICY_TEXT	VARCHAR2(4000)		Audit condition
POLICY_COLUMN	VARCHAR2(30)		Relevant column
PF_SCHEMA	VARCHAR2(30)		Owner of the audit handler function
PF_PACKAGE	VARCHAR2(30)		Name of the package containing the audit handler function
PF_FUNCTION	VARCHAR2(30)		Name of the audit handler function
ENABLED	VARCHAR2(3)		Indicates whether the policy is enabled (YES) or disabled (NO)
SEL	VARCHAR2(3)		Indicates whether the policy is applied to queries on the object (YES) or not (NO)
INS	VARCHAR2(3)		Indicates whether the policy is applied to INSERT statements on the object (YES) or not (NO)
UPD	VARCHAR2(3)		Indicates whether the policy is applied to UPDATE statements on the object (YES) or not (NO)
DEL	VARCHAR2(3)		Indicates whether the policy is applied to DELETE statements on the object (YES) or not (NO)
AUDIT_TRAIL	VARCHAR2(11)		Indicates whether the SQL_TEXT and SQL_BIND columns in the fine-grained audit trail are populated for this policy (DB_EXTENDED) or not (DB)
POLICY_COLUMN_OPTION	VARCHAR2(11)		Indicates whether all columns in the AUDIT_COLUMN parameter (ALL_COLUMNS) or any of the columns in the AUDIT_COLUMN parameter (ANY_COLUMNS) are considered for triggering fine-grained auditing

See Also:

- ["DBA_AUDIT_POLICIES"](#) on page 3-25
- ["USER_AUDIT_POLICIES"](#) on page 3-160

ALL_AUDIT_POLICY_COLUMNS

ALL_AUDIT_POLICY_COLUMNS describes the fine-grained auditing policy columns on the tables and views accessible to the current user.

Related Views

- `DBA_AUDIT_POLICY_COLUMNS` describes all fine-grained auditing policy columns in the database.
- `USER_AUDIT_POLICY_COLUMNS` describes the fine-grained auditing policy columns on the tables and views owned by the current user.

Column	Datatype	NULL	Description
<code>OBJECT_SCHEMA</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the table or view
<code>OBJECT_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the table or view
<code>POLICY_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the policy
<code>POLICY_COLUMN</code>	<code>VARCHAR2(30)</code>	NOT NULL	Relevant column of the policy

See Also:

- ["DBA_AUDIT_POLICY_COLUMNS"](#) on page 3-25
- ["USER_AUDIT_POLICY_COLUMNS"](#) on page 3-160

ALL_AW_PS

`ALL_AW_PS` describes the pagespaces in the analytic workspaces accessible to the current user.

Related Views

- `DBA_AW_PS` describes the pagespaces in all analytic workspaces in the database.
- `USER_AW_PS` describes the pagespaces in the analytic workspaces owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the analytic workspace
<code>AW_NUMBER</code>	<code>NUMBER</code>	NOT NULL	Number of the analytic workspace
<code>AW_NAME</code>	<code>VARCHAR2(30)</code>		Name of the analytic workspace
<code>PSNUMBER</code>	<code>NUMBER(10)</code>		Number of the pagespace
<code>GENERATIONS</code>	<code>NUMBER</code>		Number of active generations in the pagespace
<code>MAXPAGES</code>	<code>NUMBER</code>		Maximum pages allocated in the pagespace

See Also:

- ["DBA_AW_PS"](#) on page 3-29
- ["USER_AW_PS"](#) on page 3-161
- *Oracle OLAP Application Developer's Guide* for more information about these views

ALL_AWS

`ALL_AWS` describes the analytic workspaces accessible to the current user.

Related Views

- `DBA_AWS` describes all analytic workspaces in the database.

- `USER_AWS` describes the analytic workspaces owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the analytic workspace
<code>AW_NUMBER</code>	<code>NUMBER</code>	NOT NULL	Number of the analytic workspace
<code>AW_NAME</code>	<code>VARCHAR2(30)</code>		Name of the analytic workspace
<code>PAGESPACES</code>	<code>NUMBER</code>		Number of pagespaces in the analytic workspace
<code>GENERATIONS</code>	<code>NUMBER</code>		Number of active generations in the analytic workspace

See Also:

- ["DBA_AWS"](#) on page 3-29
- ["USER_AWS"](#) on page 3-161
- *Oracle OLAP Application Developer's Guide* for more information about these views

ALL_BASE_TABLE_MVIEWS

`ALL_BASE_TABLE_MVIEWS` describes the materialized views using materialized view logs accessible to the current user. A materialized view log can be created for a master, base table, or master materialized view. Query this view at the master site or the master materialized view site to show one row for each materialized view using a materialized view log.

Related Views

- `DBA_BASE_TABLE_MVIEWS` describes all materialized views using materialized view logs in the database.
- `USER_BASE_TABLE_MVIEWS` describes the materialized views using materialized view logs owned by the current user.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Schema in which the master table or the master materialized view was created
<code>MASTER</code>	<code>VARCHAR2(30)</code>		Name of the master table or the master materialized view
<code>MVIEW_LAST_REFRESH_TIME</code>	<code>DATE</code>		Date when the materialized view based on the master was last refreshed
<code>MVIEW_ID</code>	<code>NUMBER</code>		Unique identifier of the materialized view that is based on the master

See Also:

- ["DBA_BASE_TABLE_MVIEWS"](#) on page 3-30
- ["USER_BASE_TABLE_MVIEWS"](#) on page 3-161

ALL_CAPTURE

`ALL_CAPTURE` displays information about the capture processes that enqueue the captured changes into queues accessible to the current user.

Related View

DBA_CAPTURE displays information about all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue used for staging captured changes
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue used for staging captured changes
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set used by the capture process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set
CAPTURE_USER	VARCHAR2(30)		Current user who is enqueueing captured messages
START_SCN	NUMBER		System change number (SCN) from which the capture process will start to capture changes
STATUS	VARCHAR2(8)		Status of the capture process: <ul style="list-style-type: none"> ■ DISABLED ■ ENABLED ■ ABORTED
CAPTURED_SCN	NUMBER		System change number (SCN) of the last redo log record scanned
APPLIED_SCN	NUMBER		System change number (SCN) of the most recent message dequeued by the relevant apply processes. All changes below this SCN have been dequeued by all apply processes that apply changes captured by this capture process.
USE_DATABASE_LINK	VARCHAR2(3)		Indicates whether the source database name is used as the database link to connect to the source database from the downstream database (YES) or not (NO). If the capture process was created at the source database, then this column will be null.
FIRST_SCN	NUMBER		System change number (SCN) from which the capture process can be restarted
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database
SOURCE_DBID	NUMBER		Database ID of the source database
SOURCE_RESETLOGS_SCN	NUMBER		Resetlogs system change number (SCN) of the source database
SOURCE_RESETLOGS_TIME	NUMBER		Resetlogs time of the source database
LOGMINER_ID	NUMBER		Session ID of the LogMiner session associated with the capture process
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set used by the capture process for filtering
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set used by the capture process for filtering
MAX_CHECKPOINT_SCN	NUMBER		System change number (SCN) at which the last checkpoint was taken by the capture process
REQUIRED_CHECKPOINT_SCN	NUMBER		Lowest checkpoint SCN for which the capture process requires redo information
LOGFILE_ASSIGNMENT	VARCHAR2(8)		Logfile assignment type for the capture process: <ul style="list-style-type: none"> ■ IMPLICIT ■ EXPLICIT
STATUS_CHANGE_TIME	DATE		Time that the STATUS of the capture process was changed
ERROR_NUMBER	NUMBER		Error number if the capture process was aborted
ERROR_MESSAGE	VARCHAR2(4000)		Error message if the capture process was aborted

Column	Datatype	NULL	Description
VERSION	VARCHAR2(64)		Version number of the capture process
CAPTURE_TYPE	VARCHAR2(10)		Type of the capture process: <ul style="list-style-type: none"> ■ DOWNSTREAM ■ LOCAL

See Also: ["DBA_CAPTURE"](#) on page 3-30

ALL_CAPTURE_EXTRA_ATTRIBUTES

ALL_CAPTURE_EXTRA_ATTRIBUTES displays information about the extra attributes for the capture processes accessible to the current user.

Related View

DBA_CAPTURE_EXTRA_ATTRIBUTES displays information about the extra attributes for all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
ATTRIBUTE_NAME	VARCHAR2(30)	NOT NULL	Name of the extra attribute
INCLUDE	VARCHAR2(3)		Indicates whether the extra attribute is included (YES) or not (NO)
ROW_ATTRIBUTE	VARCHAR2(3)		Indicates whether the extra attribute is a row LCR attribute (YES) or not (NO)
DDL_ATTRIBUTE	VARCHAR2(3)		Indicates whether the extra attribute is a DDL LCR attribute (YES) or not (NO)

See Also: ["DBA_CAPTURE_EXTRA_ATTRIBUTES"](#) on page 3-30

ALL_CAPTURE_PARAMETERS

ALL_CAPTURE_PARAMETERS displays information about the parameters for the capture processes that enqueue the captured changes into queues accessible to the current user.

Related View

DBA_CAPTURE_PARAMETERS displays information about the parameters for all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
PARAMETER	VARCHAR2(128)	NOT NULL	Name of the parameter
VALUE	VARCHAR2(4000)		Parameter value
SET_BY_USER	VARCHAR2(3)		Indicates whether the parameter value was set by the user (YES) or was not set by the user (NO). If NO, then the parameter is set to its default value. If YES, then the parameter may or may not be set to its default value.

See Also: ["DBA_CAPTURE_PARAMETERS"](#) on page 3-30

ALL_CAPTURE_PREPARED_DATABASE

ALL_CAPTURE_PREPARED_DATABASE displays information about when the local database was prepared for instantiation. If the local database was not prepared for instantiation, then this view contains no rows.

Related View

DBA_CAPTURE_PREPARED_DATABASE displays information about when the local database was prepared for instantiation.

Column	Datatype	NULL	Description
TIMESTAMP	DATE		Date and time at which the local database was ready to be instantiated

See Also: ["DBA_CAPTURE_PREPARED_DATABASE"](#) on page 3-30

ALL_CAPTURE_PREPARED_SCHEMAS

ALL_CAPTURE_PREPARED_SCHEMAS displays information about the schemas prepared for instantiation that are accessible to the current user at the local database.

Related View

DBA_CAPTURE_PREPARED_SCHEMAS displays information about all schemas prepared for instantiation at the local database.

Column	Datatype	NULL	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Name of the schema that is ready to be initiated
TIMESTAMP	DATE		Date and time at which the schema was ready to be instantiated

See Also: ["DBA_CAPTURE_PREPARED_SCHEMAS"](#) on page 3-30

ALL_CAPTURE_PREPARED_TABLES

ALL_CAPTURE_PREPARED_TABLES displays information about the tables prepared for instantiation that are accessible to the current user at the local database.

Related View

DBA_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation at the local database.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table that is ready to be initiated
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table that is ready to be initiated
SCN	NUMBER	NOT NULL	Smallest system change number (SCN) for which the table can be instantiated
TIMESTAMP	DATE		Date and time at which the table was ready to be instantiated

See Also: ["DBA_CAPTURE_PREPARED_TABLES"](#) on page 3-31

ALL_CATALOG

ALL_CATALOG lists all indexes, tables, clusters, views, synonyms, and sequences accessible to the current user.

Related Views

- DBA_CATALOG lists all indexes, tables, clusters, views, synonyms, and sequences in the entire database.
- USER_CATALOG lists all indexes, tables, clusters, views, synonyms, and sequences in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the INDEX, TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the INDEX, TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED
TABLE_TYPE	VARCHAR2(11)		Type of the INDEX, TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED

ALL_CLUSTER_HASH_EXPRESSIONS

ALL_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters accessible to the current user.

Related Views

- DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database.
- USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
HASH_EXPRESSION	LONG		Text of hash function of the hash cluster

ALL_CLUSTERS

ALL_CLUSTERS describes all clusters accessible to the current user.

Related Views

- DBA_CLUSTERS describes all clusters in the database.
- USER_CLUSTERS describes all clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the cluster
PCT_FREE	NUMBER		Minimum percentage of free space in a block

Column	Datatype	NULL	Description
PCT_USED	NUMBER		Minimum percentage of used space in a block
KEY_SIZE	NUMBER		Estimated size of cluster key plus associated rows
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
AVG_BLOCKS_PER_KEY	NUMBER		Number of blocks in the table divided by number of cluster keys
CLUSTER_TYPE	VARCHAR2(5)		Type of cluster: B*-Tree index or hash
FUNCTION	VARCHAR2(15)		If a hash cluster, the hash function
HASHKEYS	NUMBER		If a hash cluster, the number of hash keys (hash buckets)
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the cluster
INSTANCES	VARCHAR2(10)		Number of instances across which the cluster is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the cluster is to be cached in the buffer cache (Y) or not (N)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the cluster
SINGLE_TABLE	VARCHAR2(5)		Indicates whether this is a single-table cluster (Y) or not (N)
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)

ALL_COL_COMMENTS

ALL_COL_COMMENTS displays comments on the columns of the tables and views accessible to the current user.

Related Views

- DBA_COL_COMMENTS displays comments on the columns of all tables and views in the database.
- USER_COL_COMMENTS displays comments on the columns of the tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
COMMENTS	VARCHAR2(4000)		Comment on the column

See Also:

- ["DBA_COL_COMMENTS"](#) on page 3-31
- ["USER_COL_COMMENTS"](#) on page 3-162

ALL_COL_PRIVS

ALL_COL_PRIVS describes the following types of grants:

- Column object grants for which the current user is the object owner, grantor, or grantee
- Column object grants for which an enabled role or PUBLIC is the grantee

Related Views

- DBA_COL_PRIVS describes all column object grants in the database.
- USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also:

- ["DBA_COL_PRIVS"](#) on page 3-31
- ["USER_COL_PRIVS"](#) on page 3-162

ALL_COL_PRIVS_MADE

ALL_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner or grantor.

Related View

USER_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column

Column	Datatype	NULL	Description
GRANTABLE	VARCHAR2 (3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: ["USER_COL_PRIVS_MADE"](#) on page 3-162

ALL_COL_PRIVS_RECD

ALL_COL_PRIVS_RECD describes the following types of grants:

- Column object grants for which the current user is the grantee
- Column object grants for which an enabled role or PUBLIC is the grantee

Related View

USER_COL_PRIVS_RECD describes the column object grants for which the current user is the grantee. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2 (30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2 (30)	NOT NUL	Owner of the object
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2 (30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2 (40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2 (3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: ["USER_COL_PRIVS_RECD"](#) on page 3-162

ALL_COLL_TYPES

ALL_COLL_TYPES describes all named collection types (varrays and nested tables) accessible to the current user.

Related Views

- DBA_COLL_TYPES describes all named collection types in the database.
- USER_COLL_TYPES describes all named collection types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the collection
TYPE_NAME	VARCHAR2 (30)	NOT NULL	Name of the collection
COLL_TYPE	VARCHAR2 (30)	NOT NULL	Description of the collection, such as VARYING ARRAY, [nested] TABLE
UPPER_BOUND	NUMBER		For varrays only, maximum size
ELEM_TYPE_MOD	VARCHAR2 (7)		Type modifier of the element
ELEM_TYPE_OWNER	VARCHAR2 (30)		Owner of the type upon which the collection is based. This value is useful primarily in the case of a user-defined type.

Column	Datatype	NULL	Description
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the datatype or user-defined type upon which the collection is based
LENGTH	NUMBER		Length of CHAR elements or maximum length of VARCHAR or VARCHAR2 elements
PRECISION	NUMBER		Decimal precision of NUMBER or DECIMAL elements; binary precision of FLOAT elements
SCALE	NUMBER		Scale of NUMBER or DECIMAL elements
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set (CHAR_CS NCHAR_CS)
ELEM_STORAGE	VARCHAR2(7)		Obsolete column
NULLS_STORED	VARCHAR2(3)		Obsolete column

ALL_CONS_COLUMNS

ALL_CONS_COLUMNS describes columns that are accessible to the current user and that are specified in constraints.

Related Views

- DBA_CONS_COLUMNS describes all columns in the database that are specified in constraints.
- USER_CONS_COLUMNS describes columns that are owned by the current user and that are specified in constraints.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name of the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name of the column or attribute of the object type column specified in the constraint definition Note: If you create a constraint on a user-defined REF column, the system creates the constraint on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: "REF_name"."attribute"
POSITION	NUMBER		Original position of column or attribute in the definition of the object

ALL_CONS_OBJ_COLUMNS

ALL_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables accessible to the current user.

Related Views

- DBA_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in all tables in the database.

- `USER_CONS_OBJ_COLUMNS` displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the table
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table containing the object column or attribute
<code>COLUMN_NAME</code>	<code>VARCHAR2(4000)</code>		Fully qualified name of the object column or attribute
<code>CONS_TYPE_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the type that the column (or element) is constrained to
<code>CONS_TYPE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the type that the column (or element) is constrained to
<code>CONS_TYPE_ONLY</code>	<code>VARCHAR2(15)</code>		Indicates whether the column (or element) is constrained to <code>ONLY</code> type (Y) or not (N)

See Also:

- ["DBA_CONS_OBJ_COLUMNS"](#) on page 3-34
- ["USER_CONS_OBJ_COLUMNS"](#) on page 3-162

ALL_CONSTRAINTS

`ALL_CONSTRAINTS` describes constraint definitions on tables accessible to the current user.

Related Views

- `DBA_CONSTRAINTS` describes all constraint definitions in the database.
- `USER_CONSTRAINTS` describes constraint definitions on tables in the current user's schema.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the constraint definition
<code>CONSTRAINT_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the constraint definition
<code>CONSTRAINT_TYPE</code>	<code>VARCHAR2(1)</code>		Type of constraint definition: <ul style="list-style-type: none"> ■ C (check constraint on a table) ■ P (primary key) ■ U (unique key) ■ R (referential integrity) ■ V (with check option, on a view) ■ O (with read only, on a view)
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name associated with the table (or view) with constraint definition
<code>SEARCH_CONDITION</code>	LONG		Text of search condition for a check constraint
<code>R_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of table referred to in a referential constraint
<code>R_CONSTRAINT_NAME</code>	<code>VARCHAR2(30)</code>		Name of the unique constraint definition for referenced table
<code>DELETE_RULE</code>	<code>VARCHAR2(9)</code>		Delete rule for a referential constraint (CASCADE or NO ACTION)
<code>STATUS</code>	<code>VARCHAR2(8)</code>		Enforcement status of constraint (ENABLED or DISABLED)

Column	Datatype	NULL	Description
DEFERRABLE	VARCHAR2(14)		Whether the constraint is deferrable
DEFERRED	VARCHAR2(9)		Whether the constraint was initially deferred
VALIDATED	VARCHAR2(13)		Whether all data obeys the constraint (VALIDATED or NOT VALIDATED)
GENERATED	VARCHAR2(14)		Whether the name of the constraint is user or system generated
BAD	VARCHAR2(3)		A YES value indicates that this constraint specifies a century in an ambiguous manner. To avoid errors resulting from this ambiguity, rewrite the constraint using the TO_DATE function with a four-digit year. See Also: the TO_DATE function in <i>Oracle Database SQL Reference</i> and <i>Oracle Database Application Developer's Guide - Fundamentals</i>
RELY	VARCHAR2(4)		Whether an enabled constraint is enforced or unenforced. See Also: the <i>constraints</i> in <i>Oracle Database SQL Reference</i>
LAST_CHANGE	DATE		When the constraint was last enabled or disabled
INDEX_OWNER	VARCHAR2(30)		Name of the user owning the index
INDEX_NAME	VARCHAR2(30)		Name of the index
INVALID	VARCHAR2(7)		Whether the constraint is invalid
VIEW_ONLY	VARCHAR2(14)		Whether the constraint depends on a view

ALL_CONTEXT

ALL_CONTEXT describes all context namespaces in the current session for which attributes and values have been specified using the DBMS_SESSION.SET_CONTEXT procedure. This view does not display the TYPE column.

Related View

DBA_CONTEXT describes all context namespaces defined in the database, regardless whether any attributes have been specified for them using the DBMS_SESSION.SET_CONTEXT procedure.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	Name of the context namespace
SCHEMA	VARCHAR2(30)	NOT NULL	Schema name of the designated package that can set attributes using this namespace
PACKAGE	VARCHAR2(30)	NOT NULL	Package name of the designated package that can set attributes using this namespace
TYPE	VARCHAR2(22)		Type of the context: <ul style="list-style-type: none"> ■ ACCESSED LOCALLY ■ INITIALIZED EXTERNALLY ■ ACCESSED GLOBALLY ■ INITIALIZED GLOBALLY

ALL_DB_LINKS

ALL_DB_LINKS describes the database links accessible to the current user. This view does not display the PASSWORD column.

Related Views

- `DBA_DB_LINKS` describes all database links in the database. This view does not display the `PASSWORD` column.
- `USER_DB_LINKS` describes the database links owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the database link
<code>DB_LINK</code>	<code>VARCHAR2(128)</code>	NOT NULL	Name of the database link
<code>USERNAME</code>	<code>VARCHAR2(30)</code>		Name of the user when logging in
<code>PASSWORD</code>	<code>VARCHAR2(30)</code>		Login password for the current user
<code>HOST</code>	<code>VARCHAR2(2000)</code>		Oracle Net connect string
<code>CREATED</code>	<code>DATE</code>	NOT NULL	Creation time of the database link

See Also:

- ["DBA_DB_LINKS"](#) on page 3-35
- ["USER_DB_LINKS"](#) on page 3-163

ALL_DEF_AUDIT_OPTS

`ALL_DEF_AUDIT_OPTS` contains default object-auditing options that will be applied when objects are created. The output for each column takes one of the following forms:

- `-/-`: no default auditing
- `S/-`: auditing whenever successful
- `-/S`: auditing whenever not successful

Column	Datatype	NULL	Description
<code>ALT</code>	<code>VARCHAR2(3)</code>		Auditing ALTER WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>AUD</code>	<code>VARCHAR2(3)</code>		Auditing AUDIT WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>COM</code>	<code>VARCHAR2(3)</code>		Auditing COMMENT WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>DEL</code>	<code>VARCHAR2(3)</code>		Auditing DELETE WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>GRA</code>	<code>VARCHAR2(3)</code>		Auditing GRANT WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>IND</code>	<code>VARCHAR2(3)</code>		Auditing INDEX WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>INS</code>	<code>VARCHAR2(3)</code>		Auditing INSERT WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>LOC</code>	<code>VARCHAR2(3)</code>		Auditing LOCK WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>REN</code>	<code>VARCHAR2(3)</code>		Auditing RENAME WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>SEL</code>	<code>VARCHAR2(3)</code>		Auditing SELECT WHENEVER SUCCESSFUL / UNSUCCESSFUL
<code>UPD</code>	<code>VARCHAR2(3)</code>		Auditing UPDATE WHENEVER SUCCESSFUL / UNSUCCESSFUL

Column	Datatype	NULL	Description
REF	CHAR(3)		Auditing REFERENCES WHENEVER SUCCESSFUL / UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL / UNSUCCESSFUL
FBK	VARCHAR2(3)		Auditing FLASHBACK WHENEVER SUCCESSFUL / UNSUCCESSFUL

ALL_DEPENDENCIES

ALL_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers accessible to the current user, including dependencies on views created without any database links. This view does not display the SCHEMAID column.

Related Views

- DBA_DEPENDENCIES describes all dependencies between objects in the database. This view does not display the SCHEMAID column.
- USER_DEPENDENCIES describes dependencies between objects in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(17)		Type of object
REFERENCED_OWNER	VARCHAR2(30)		Owner of the parent object
REFERENCED_NAME	VARCHAR2(64)		Type of parent object
REFERENCED_TYPE	VARCHAR2(17)		Type of referenced object
REFERENCED_LINK_NAME	VARCHAR2(128)		Name of the link to the parent object (if remote)
SCHEMAID	NUMBER		ID of the current schema
DEPENDENCY_TYPE	VARCHAR2(4)		Whether the dependency is a REF dependency (REF) or not (HARD)

ALL_DIM_ATTRIBUTES

ALL_DIM_ATTRIBUTES describes the relationship between a dimension level and a functionally dependent column. The level columns and the dependent column must be in the same table.

Related Views

- DBA_DIM_ATTRIBUTES describes all such dimension relationships in the database.
- USER_DIM_ATTRIBUTES describes all such dimension attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
ATTRIBUTE_NAME	VARCHAR2(30)		Name of the attribute

Column	Datatype	NULL	Description
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Dependent column name
INFERRED	CHAR(1)		Indicates whether the attribute is inferred from a JOIN KEY specification (Y) or not (N)

ALL_DIM_CHILD_OF

ALL_DIM_CHILD_OF describes hierarchical relationships of 1 to n between the pairs of levels in the dimensions accessible to the current user.

Related Views

- DBA_DIM_CHILD_OF describes all such hierarchical relationships in the database.
- USER_DIM_CHILD_OF describes all such hierarchical attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name
POSITION	NUMBER	NOT NULL	Hierarchical position within this hierarchy, position 1 being the most detailed
CHILD_LEVEL_NAME	VARCHAR2(30)		Child side of 1:n relationship
JOIN_KEY_ID	VARCHAR2(40)		If non-null, then the child joins to the parent
PARENT_LEVEL_NAME	VARCHAR2(30)		Parent side of 1:n relationship in relation to the CHILD_LEVEL_NAME

ALL_DIM_HIERARCHIES

ALL_DIM_HIERARCHIES describes all dimension hierarchies accessible to the current user.

Related Views

- DBA_DIM_HIERARCHIES describes all such hierarchies in the database.
- USER_DIM_HIERARCHIES describes all such hierarchies owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name

ALL_DIM_JOIN_KEY

ALL_DIM_JOIN_KEY describes the joins between two dimension tables that are accessible to the current user. The join is always specified between a parent dimension level column and a child column.

Related Views

- DBA_DIM_JOIN_KEY describes all such joins in the database.
- USER_DIM_JOIN_KEY describes all such joins owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
DIM_KEY_ID	NUMBER	NOT NULL	Join key ID (unique within a dimension)
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
HIERARCHY_NAME	VARCHAR2(30)		Name of the hierarchy
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column

ALL_DIM_LEVEL_KEY

ALL_DIM_LEVEL_KEY describes a column of a dimension level accessible to the current user. The position of a column within a level is specified by KEY_POSITION.

Related Views

- DBA_DIM_LEVEL_KEY describes all columns of dimension levels in the database.
- USER_DIM_LEVEL_KEY describes all columns of dimension levels owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

ALL_DIM_LEVELS

ALL_DIM_LEVELS describes the dimension levels accessible to the current user. All columns of a dimension level must come from the same relation.

Related Views

- DBA_DIM_LEVELS describes all dimension levels in the database.
- USER_DIM_LEVELS describes the levels of all dimensions owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Unique within a dimension
NUM_COLUMNS	NUMBER		Number of columns in the level definition
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Owner of the detail object that the keys of this level come from

Column	Datatype	NULL	Description
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Name of the table that the keys of this level come from

ALL_DIMENSIONS

ALL_DIMENSIONS describes dimension objects accessible to the current user.

Related Views

- DBA_DIMENSIONS describes all dimensions in the database.
- USER_DIMENSIONS describes dimensions in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
INVALID	VARCHAR2(1)		Indicates whether the dimension is invalid (Y) or valid (N)
COMPILE_STATE	VARCHAR2(13)		Compile status of the dimension: <ul style="list-style-type: none"> ■ INVALID ■ NEEDS_COMPILE ■ ERROR
REVISION	NUMBER		Dimension revision level

ALL_DIRECTORIES

ALL_DIRECTORIES describes all directories accessible to the current user.

Related View

DBA_DIRECTORIES describes all directories in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the directory (always SYS)
DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the directory
DIRECTORY_PATH	VARCHAR2(4000)		Operating system pathname for the directory

ALL_ERRORS

ALL_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) accessible to the current user.

Related Views

- DBA_ERRORS describes all current errors on all views, procedures, functions, packages, and package bodies in the database.
- USER_ERRORS describes all current errors on all views, procedures, functions, packages, and package bodies owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: <ul style="list-style-type: none"> ■ VIEW ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY ■ LIBRARY ■ JAVA SOURCE ■ JAVA CLASS ■ DIMENSION
SEQUENCE	NUMBER	NOT NULL	Sequence number (for ordering purposes)
LINE	NUMBER	NOT NULL	Line number at which the error occurred
POSITION	NUMBER	NOT NULL	Position in the line at which the error occurred
TEXT	VARCHAR2(4000)	NOT NULL	Text of the error
ATTRIBUTE	VARCHAR2(9)		Indicates whether the selected row is an error (ERROR) or a warning (WARNING)
MESSAGE_NUMBER	NUMBER		Numerical error number (without any prefix)

ALL_EVALUATION_CONTEXT_TABLES

ALL_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_TABLES describes the tables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
TABLE_ALIAS	VARCHAR2(30)		Alias for a table in the evaluation context
TABLE_NAME	VARCHAR2(4000)		Name of the table referred to by the table alias

See Also:

- ["DBA_EVALUATION_CONTEXT_TABLES"](#) on page 3-39
- ["USER_EVALUATION_CONTEXT_TABLES"](#) on page 3-164

ALL_EVALUATION_CONTEXT_VARS

ALL_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_VARS describes the variables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
VARIABLE_NAME	VARCHAR2(30)		Name of a variable in the evaluation context
VARIABLE_TYPE	VARCHAR2(4000)		Datatype of the variable
VARIABLE_VALUE_FUNCTION	VARCHAR2(4000)		Function used to retrieve the value of the variable (NULL for variables that are not implicit)
VARIABLE_METHOD_FUNCTION	VARCHAR2(228)		Function used to retrieve the result of method invocation on the variable. Such a function can speed up evaluation, if there are many simple rules that invoke the method on the variable.

See Also:

- ["DBA_EVALUATION_CONTEXT_VARS"](#) on page 3-39
- ["USER_EVALUATION_CONTEXT_VARS"](#) on page 3-164

ALL_EVALUATION_CONTEXTS

ALL_EVALUATION_CONTEXTS describes the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXTS describes all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXTS describes the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
EVALUATION_FUNCTION	VARCHAR2(4000)		Evaluation function associated with the evaluation context, if any
EVALUATION_CONTEXT_COMMENT	VARCHAR2(4000)		Comment specified with the evaluation context, if any

See Also:

- ["DBA_EVALUATION_CONTEXTS"](#) on page 3-39
- ["USER_EVALUATION_CONTEXTS"](#) on page 3-164

ALL_EXTERNAL_LOCATIONS

ALL_EXTERNAL_LOCATIONS describes the locations (data sources) of the external tables accessible to the current user.

Related Views

- DBA_EXTERNAL_LOCATIONS describes the locations (data sources) of all external tables in the database.
- USER_EXTERNAL_LOCATIONS describes the locations (data sources) of the external tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the external table location
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the corresponding external table
LOCATION	VARCHAR2(4000)		External table location clause
DIRECTORY_OWNER	CHAR(3)		Owner of the directory containing the external table location
DIRECTORY_NAME	VARCHAR2(30)		Name of the directory containing the external table location

See Also:

- ["DBA_EXTERNAL_LOCATIONS"](#) on page 3-41
- ["USER_EXTERNAL_LOCATIONS"](#) on page 3-165

ALL_EXTERNAL_TABLES

ALL_EXTERNAL_TABLES describes the external tables accessible to the current user.

Related Views

- DBA_EXTERNAL_TABLES describes all external tables in the database.
- USER_EXTERNAL_TABLES describes the external tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the external table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the external table
TYPE_OWNER	CHAR(3)		Owner of the implementation type for the external table access driver
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the implementation type for the external table access driver
DEFAULT_DIRECTORY_OWNER	CHAR(3)		Owner of the default directory for the external table

Column	Datatype	NULL	Description
DEFAULT_DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the default directory for the external table
REJECT_LIMIT	VARCHAR2(40)		Reject limit for the external table
ACCESS_TYPE	VARCHAR2(7)		Type of access parameters for the external table: <ul style="list-style-type: none"> ▪ BLOB ▪ CLOB
ACCESS_PARAMETERS	VARCHAR2(4000)		Access parameters for the external table
PROPERTY	VARCHAR2(10)		Property of the external table: <ul style="list-style-type: none"> ▪ REFERENCED - Referenced columns ▪ ALL - All columns

See Also:

- ["DBA_EXTERNAL_TABLES"](#) on page 3-41
- ["USER_EXTERNAL_TABLES"](#) on page 3-165

ALL_IND_COLUMNS

ALL_IND_COLUMNS describes the columns of indexes on all tables accessible to the current user.

Note: For join indexes, the TABLE_NAME and TABLE_OWNER columns in this view may not match the TABLE_NAME and TABLE_OWNER columns you find in the *_INDEXES (and other similar) data dictionary views.

Related Views

- DBA_IND_COLUMNS describes the columns of indexes on all tables in the database.
- USER_IND_COLUMNS describes the columns of indexes owned by the current user and columns of indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER columns.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table or cluster
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of object type column Note: If you create an index on a user-defined REF column, the system creates the index on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: "REF_name"."attribute"
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within the index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column
CHAR_LENGTH	NUMBER		Maximum codepoint length of the column

Column	Datatype	NULL	Description
DESCEND	VARCHAR2(4)		Whether the column is sorted in descending order (Y/N)

ALL_IND_EXPRESSIONS

ALL_IND_EXPRESSIONS describes the expressions of function-based indexes on tables accessible to the current user.

Related Views

- DBA_IND_EXPRESSIONS describes the expressions of all function-based indexes in the database.
- USER_IND_EXPRESSIONS describes the expressions of function-based indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER columns.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Function-based index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

ALL_IND_PARTITIONS

ALL_IND_PARTITIONS describes, for each index partition accessible to the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements.

Related Views

- DBA_IND_PARTITIONS describes all index partitions in the database.
- USER_IND_PARTITIONS describes the index partitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	Description
INDEX_OWNER	VARCHAR2(30)	Owner of the index
INDEX_NAME	VARCHAR2(30)	Name of the index
COMPOSITE	VARCHAR2(3)	Indicates whether the partition belongs to a local index on a composite-partitioned table (YES) or not (NO)
PARTITION_NAME	VARCHAR2(30)	Name of the partition
SUBPARTITION_COUNT	NUMBER	If a local index on a composite-partitioned table, the number of subpartitions in the partition
HIGH_VALUE	LONG	Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	Length of partition bound value expression
PARTITION_POSITION	NUMBER	Position of the partition within the index
STATUS	VARCHAR2(8)	Whether index partition is USABLE or UNUSABLE
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace containing the partition
PCT_FREE	NUMBER	Minimum percentage of free space in a block

Column	Datatype	Description
INI_TRANS	NUMBER	Initial number of transactions
MAX_TRANS	NUMBER	Maximum number of transactions
INITIAL_EXTENT	NUMBER	Size of the initial extent in bytes
NEXT_EXTENT	NUMBER	Size of secondary extents in bytes
MIN_EXTENT	NUMBER	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	Percentage increase in extent size
FREELISTS	NUMBER	Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER	Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(7)	Logging attribute of the partition
COMPRESSION	VARCHAR2(8)	Indicates whether key compression is enabled for a partitioned index (ENABLED) or not (DISABLED); null for a nonpartitioned index.
BLEVEL	NUMBER	B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER	Number of leaf blocks in the index partition
DISTINCT_KEYS	NUMBER	Number of distinct keys in the index partition
AVG_LEAF_BLOCKS_PER_KEY	NUMBER	Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER	Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR	NUMBER	Indicates the amount of order of the rows in the table based on the values of the index. <ul style="list-style-type: none"> ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER	Number of rows returned by the ANALYZE statement
SAMPLE_SIZE	NUMBER	Sample size used in analyzing this partition
LAST_ANALYZED	DATE	Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)	Actual buffer pool for the partition
USER_STATS	VARCHAR2(3)	Indicates whether statistics were entered directly by the user (YES) or not (NO)
PCT_DIRECT_ACCESS	NUMBER	If a secondary index on index-organized table, the percentage of rows with VALID guess
GLOBAL_STATS	VARCHAR2(3)	Indicates whether statistics for the partition were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
DOMIDX_OPSTATUS	VARCHAR2(6)	
PARAMETERS	VARCHAR2(1000)	

ALL_IND_STATISTICS

ALL_IND_STATISTICS displays optimizer statistics for the indexes on the tables accessible to the current user.

Related Views

- `DBA_IND_STATISTICS` displays optimizer statistics for all indexes in the database.
- `USER_IND_STATISTICS` displays optimizer statistics for the indexes on the tables owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the index
<code>INDEX_NAME</code>	<code>VARCHAR2(30)</code>		Name of the index
<code>PARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Name of the partition
<code>PARTITION_POSITION</code>	<code>NUMBER</code>		Position of the partition within the index
<code>SUBPARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Name of the subpartition
<code>SUBPARTITION_POSITION</code>	<code>NUMBER</code>		Position of the subpartition within the partition
<code>OBJECT_TYPE</code>	<code>VARCHAR2(12)</code>		Type of the object: <ul style="list-style-type: none"> ■ <code>INDEX</code> ■ <code>PARTITION</code> ■ <code>SUBPARTITION</code>
<code>BLEVEL</code>	<code>NUMBER</code>		B-Tree level
<code>LEAF_BLOCKS</code>	<code>NUMBER</code>		Number of leaf blocks in the index
<code>DISTINCT_KEYS</code>	<code>NUMBER</code>		Number of distinct keys in the index
<code>AVG_LEAF_BLOCKS_PER_KEY</code>	<code>NUMBER</code>		Average number of leaf blocks per key
<code>AVG_DATA_BLOCKS_PER_KEY</code>	<code>NUMBER</code>		Average number of data blocks per key
<code>CLUSTERING_FACTOR</code>	<code>NUMBER</code>		Indicates the amount of order of the rows in the table based on the values of the index. <ul style="list-style-type: none"> ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
<code>NUM_ROWS</code>	<code>NUMBER</code>		Number of rows in the index
<code>AVG_CACHED_BLOCKS</code>	<code>NUMBER</code>		Average number of blocks in the buffer cache
<code>AVG_CACHE_HIT_RATIO</code>	<code>NUMBER</code>		Average cache hit ratio for the object
<code>SAMPLE_SIZE</code>	<code>NUMBER</code>		Sample size used in analyzing the index
<code>LAST_ANALYZED</code>	<code>DATE</code>		Date of the most recent time the index was analyzed
<code>GLOBAL_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether statistics were calculated without merging underlying partitions (<code>YES</code>) or not (<code>NO</code>)
<code>USER_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether statistics were entered directly by the user (<code>YES</code>) or not (<code>NO</code>)

See Also:

- ["DBA_IND_STATISTICS"](#) on page 3-82
- ["USER_IND_STATISTICS"](#) on page 3-165

ALL_IND_SUBPARTITIONS

ALL_IND_SUBPARTITIONS describes, for each index subpartition accessible to the current user, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements.

Related Views

- DBA_IND_SUBPARTITIONS describes all index subpartitions in the database.
- USER_IND_SUBPARTITIONS describes the index subpartitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
STATUS	VARCHAR2(8)		Whether index partition is usable or not
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index partition
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.

Column	Datatype	NULL	Description
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. <ul style="list-style-type: none"> If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER		Number of rows in this index subpartition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this subpartition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the subpartition
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition statistics were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO)

ALL_INDEXES

ALL_INDEXES describes the indexes on the tables accessible to the current user. To gather statistics for this view and the related views DBA_INDEXES and USER_INDEXES, use the SQL ANALYZE statement.

Related Views

- DBA_INDEXES describes all indexes in the database.
- USER_INDEXES describes the indexes owned by the current user. This view does not display the OWNER column.

Note: Column names followed by an asterisk are populated only if you collect statistics on the index using the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(27)		Type of the index: <ul style="list-style-type: none"> NORMAL BITMAP FUNCTION-BASED NORMAL FUNCTION-BASED BITMAP DOMAIN
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the indexed object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
TABLE_TYPE	CHAR(5)		Type of the indexed object (for example, TABLE, CLUSTER)
UNIQUENESS	VARCHAR2(9)		Indicates whether the index is UNIQUE or NONUNIQUE

Column	Datatype	NULL	Description
COMPRESSION	VARCHAR2 (8)		Indicates whether index compression is enabled (ENABLED) or not (DISABLED)
PREFIX_LENGTH	NUMBER		Number of columns in the prefix of the compression key
TABLESPACE_NAME	VARCHAR2 (30)		Name of the tablespace containing the index
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent
NEXT_EXTENT	NUMBER		Size of secondary extents
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		Column ID of the last column to be included in index-organized table primary key (non-overflow) index. This column maps to the COLUMN_ID column of the *_TAB_COLUMNS data dictionary views.
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
PCT_FREE	NUMBER		Minimum percentage of free space in a block
LOGGING	VARCHAR2 (3)		Logging information
BLEVEL*	NUMBER		B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS*	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS*	NUMBER		Number of distinct indexed values. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table (USER_TABLES.NUM_ROWS)
AVG_LEAF_BLOCKS_PER_KEY*	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY*	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR*	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. <ul style="list-style-type: none"> ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks. For bitmap indexes, this column is not applicable and is not used.
STATUS	VARCHAR2 (8)		Indicates whether a nonpartitioned index is VALID or UNUSABLE
NUM_ROWS	NUMBER		Number of rows in the index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze the index

Column	Datatype	NULL	Description
LAST_ANALYZED	DATE		Date on which this index was most recently analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index
INSTANCES	VARCHAR2(40)		Number of instances across which the indexes to be scanned
PARTITIONED	VARCHAR2(3)		Indicates whether the index is partitioned (YES) or not (NO)
TEMPORARY	VARCHAR2(1)		Indicates whether the index is on a temporary table
GENERATED	VARCHAR2(1)		Indicates whether the name of the index is system generated (Y N)
SECONDARY	VARCHAR2(1)		Indicates whether the index is a secondary object created by the <code>ODCIIndexCreate</code> method of the Oracle Data Cartridge (Y N)
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool to be used for the index blocks
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: <ul style="list-style-type: none"> ■ <code>SYS\$SESSION</code> - Rows are preserved for the duration of the session ■ <code>SYS\$TRANSACTION</code> - Rows are deleted after COMMIT Null for a permanent table
PCT_DIRECT_ACCESS	NUMBER		For a secondary index on an index-organized table, the percentage of rows with <code>VALID</code> guess
ITYP_OWNER	VARCHAR2(30)		For a domain index, the owner of the indextype
ITYP_NAME	VARCHAR2(30)		For a domain index, the name of the indextype
PARAMETERS	VARCHAR2(1000)		For a domain index, the parameter string
GLOBAL_STATS	VARCHAR2(3)		For partitioned indexes, indicates whether statistics were collected by analyzing the index as a whole (YES) or were estimated from statistics on underlying index partitions and subpartitions (NO)
DOMIDX_STATUS	VARCHAR2(12)		Status of the domain index: <ul style="list-style-type: none"> ■ <code>NULL</code> - Index is not a domain index ■ <code>VALID</code> - Index is a valid domain index ■ <code>IDXTYP_INVLD</code> - Indextype of the domain index is invalid
DOMIDX_OPSTATUS	VARCHAR2(6)		Status of the operation on the domain index: <ul style="list-style-type: none"> ■ <code>NULL</code> - Index is not a domain index ■ <code>VALID</code> - Operation performed without errors ■ <code>FAILED</code> - Operation failed with an error
FUNCIDX_STATUS	VARCHAR2(8)		Status of a function-based index: <ul style="list-style-type: none"> ■ <code>NULL</code> - Index is not a function-based index ■ <code>ENABLED</code> - Function-based index is enabled ■ <code>DISABLED</code> - Function-based index is disabled
JOIN_INDEX	VARCHAR2(3)		Indicates whether the index is a join index (YES) or not (NO)
IOT_REDUNDANT_PKEY_ELIM	VARCHAR2(3)		Indicates whether redundant primary key columns are eliminated from secondary indexes on index-organized tables (YES) or not (NO)
DROPPED	VARCHAR2(3)		Indicates whether the index has been dropped and is in the recycle bin (YES) or not (NO); null for partitioned tables

ALL_INDEXTYPE_ARRAYTYPES

ALL_INDEXTYPE_ARRAYTYPES displays information about the array types specified by the indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_ARRAYTYPES displays information about the array types specified by all indextypes in the database.
- USER_INDEXTYPE_ARRAYTYPES displays information about the array types specified by the indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
BASE_TYPE_SCHEMA	VARCHAR2(30)		Name of the base type schema
BASE_TYPE_NAME	VARCHAR2(30)		Name of the base type name
BASE_TYPE	VARCHAR2(30)		Datatype of the base type
ARRAY_TYPE_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the array type schema
ARRAY_TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the array type name

See Also:

- ["DBA_INDEXTYPE_ARRAYTYPES"](#) on page 3-82
- ["USER_INDEXTYPE_ARRAYTYPES"](#) on page 3-166

ALL_INDEXTYPE_COMMENTS

ALL_INDEXTYPE_COMMENTS displays comments for the user-defined indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_COMMENTS displays comments for all user-defined indextypes in the database.
- USER_INDEXTYPE_COMMENTS displays comments for the user-defined indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the user-defined indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the user-defined indextype
COMMENTS	VARCHAR2(4000)		Comment for the user-defined indextype

See Also:

- ["DBA_INDEXTYPE_COMMENTS"](#) on page 3-82
- ["USER_INDEXTYPE_COMMENTS"](#) on page 3-166

ALL_INDEXTYPE_OPERATORS

ALL_INDEXTYPE_OPERATORS lists all operators supported by indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_OPERATORS lists all operators supported by indextypes in the database.
- USER_INDEXTYPE_OPERATORS lists all operators supported by indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
OPERATOR_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the operator schema
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator for which the indextype is defined
BINDING#	NUMBER	NOT NULL	Binding# associated with the operator

See Also:

- ["DBA_INDEXTYPE_OPERATORS"](#) on page 3-82
- ["USER_INDEXTYPE_OPERATORS"](#) on page 3-166

ALL_INDEXTYPES

ALL_INDEXTYPES displays information about the indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPES displays information about all indextypes in the database.
- USER_INDEXTYPES displays information about the indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
IMPLEMENTATION_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the schema for the indextype implementation (that is, containing the indextype operators)
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype implementation type
INTERFACE_VERSION	NUMBER		Version of the indextype interface
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of the indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype
PARTITIONING	VARCHAR2(10)		Kinds of local partitioning supported by the indextype: <ul style="list-style-type: none"> ■ NONE ■ RANGE ■ HASH ■ HASH, RANGE

Column	Datatype	NULL	Description
ARRAY_DML	VARCHAR2(3)		Indicates whether the indextype supports array DML (YES) or not (NO)

See Also:

- ["DBA_INDEXTYPES"](#) on page 3-82
- ["USER_INDEXTYPES"](#) on page 3-166

ALL_INTERNAL_TRIGGERS

ALL_INTERNAL_TRIGGERS describes internal triggers on tables accessible to the current user. Internal triggers are internal pieces of code executed when a particular flag is set for a table. This view does not display the OWNER_NAME column.

Related Views

- DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database.
- USER_INTERNAL_TRIGGERS describes all internal triggers on tables owned by the current user. This view does not display the OWNER_NAME column.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the table on which the trigger is defined
OWNER_NAME	VARCHAR2(30)		Owner of the table
INTERNAL_TRIGGER_TYPE	VARCHAR2(19)		Indicates the type of internal trigger on the table

ALL_JAVA_ARGUMENTS

ALL_JAVA_ARGUMENTS displays argument information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_ARGUMENTS displays argument information about all stored Java classes in the database.
- USER_JAVA_ARGUMENTS displays argument information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the hosting method of the argument
METHOD_NAME	VARCHAR2(4000)		Name of the hosting method of the argument
ARGUMENT_POSITION	NUMBER		Position of the argument, starting from 0
ARRAY_DEPTH	NUMBER		Array depth of the type of the argument

Column	Datatype	NULL	Description
BASE_TYPE	VARCHAR2(7)		Base type of the type of the argument: <ul style="list-style-type: none"> ▪ int ▪ long ▪ float ▪ double ▪ boolean ▪ byte ▪ char ▪ short ▪ class
ARGUMENT_CLASS	VARCHAR2(4000)		Actual class name of the argument if the base type is class

See Also:

- ["DBA_JAVA_ARGUMENTS"](#) on page 3-83
- ["USER_JAVA_ARGUMENTS"](#) on page 3-166

ALL_JAVA_CLASSES

ALL_JAVA_CLASSES displays class level information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_CLASSES displays class level information about all stored Java classes in the database.
- USER_JAVA_CLASSES displays class level information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
MAJOR	NUMBER		Major version number of the Java class, as defined in the JVM specification
MINOR	NUMBER		Minor version number of the Java class, as defined in the JVM specification
KIND	VARCHAR2(9)		Indicates whether the stored object is a Java class (CLASS) or a Java interface (INTERFACE)
ACCESSIBILITY	VARCHAR2(6)		Accessibility of the Java class
IS_INNER	VARCHAR2(3)		Indicates whether this Java class is an inner class (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether this Java class is an abstract class (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether this Java class is a final class (YES) or not (NO)
IS_DEBUG	VARCHAR2(3)		Indicates whether this Java class contains debug information (YES) or not (NO)
SOURCE	VARCHAR2(4000)		Source designation of the Java class
SUPER	VARCHAR2(4000)		Super class of this Java class

Column	Datatype	NULL	Description
OUTER	VARCHAR2(4000)		Outer class of this Java class if this Java class is an inner class

See Also:

- ["DBA_JAVA_CLASSES"](#) on page 3-83
- ["USER_JAVA_CLASSES"](#) on page 3-167

ALL_JAVA_DERIVATIONS

ALL_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes accessible to the current user.

Related Views

- DBA_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for all Java classes in the database.
- USER_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java source object
SOURCE_NAME	VARCHAR2(4000)		Name of the Java source object
CLASS_INDEX	NUMBER		Index of the derived Java class object
CLASS_NAME	VARCHAR2(4000)		Name of the derived Java class object
RESOURCE_INDEX	NUMBER		Index of the derived Java resource object
RESOURCE_NAME	VARCHAR2(4000)		Name of the derived Java resource object

See Also:

- ["DBA_JAVA_DERIVATIONS"](#) on page 3-83
- ["USER_JAVA_DERIVATIONS"](#) on page 3-167

ALL_JAVA_FIELDS

ALL_JAVA_FIELDS displays field information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_FIELDS displays field information about all stored Java classes in the database.
- USER_JAVA_FIELDS displays field information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class

Column	Datatype	NULL	Description
NAME	VARCHAR2(4000)		Name of the Java class
FIELD_INDEX	NUMBER		Index of the field
FIELD_NAME	VARCHAR2(4000)		Name of the field identified by the FIELD_INDEX column
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the field: <ul style="list-style-type: none"> ▪ PUBLIC ▪ PRIVATE ▪ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the field is a static field (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the field is a final field (YES) or not (NO)
IS_VOLATILE	VARCHAR2(3)		Indicates whether the field is volatile (YES) or not (NO)
IS_TRANSIENT	VARCHAR2(3)		Indicates whether the field is transient (YES) or not (NO)
ARRAY_DEPTH	NUMBER		Array depth of the type of the field
BASE_TYPE	VARCHAR2(7)		Base type of the type of the field: <ul style="list-style-type: none"> ▪ int ▪ long ▪ float ▪ double ▪ boolean ▪ byte ▪ char ▪ short ▪ class
FIELD_CLASS	VARCHAR2(4000)		Actual class name of the base object if the base type is class

See Also:

- ["DBA_JAVA_FIELDS"](#) on page 3-83
- ["USER_JAVA_FIELDS"](#) on page 3-167

ALL_JAVA_IMPLMENTS

ALL_JAVA_IMPLMENTS describes interfaces implemented by the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_IMPLMENTS describes interfaces implemented by all stored Java classes in the database.
- USER_JAVA_IMPLMENTS describes interfaces implemented by the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INTERFACE_INDEX	NUMBER		Index of the interfaces implemented by the Java class

Column	Datatype	NULL	Description
INTERFACE_NAME	VARCHAR2(4000)		Name of the interface identified by the INTERFACE_INDEX column

See Also:

- ["DBA_JAVA_IMPLMENTS"](#) on page 3-83
- ["USER_JAVA_IMPLMENTS"](#) on page 3-167

ALL_JAVA_INNERS

ALL_JAVA_INNERS displays information about inner classes referred to by the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_INNERS displays information about inner classes referred to by all stored Java classes in the database.
- USER_JAVA_INNERS displays information about inner classes referred to by the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INNER_INDEX	NUMBER		Index of the referred inner class
SIMPLE_NAME	VARCHAR2(4000)		Simple name of the referred inner class
FULL_NAME	VARCHAR2(4000)		Full name of the referred inner class
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the referred inner class: <ul style="list-style-type: none"> ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the referred inner class is declared static in the source file (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the referred inner class is declared final in the source file (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether the referred inner class is declared abstract in the source file (YES) or not (NO)
IS_INTERFACE	VARCHAR2(3)		Indicates whether the referred inner class is declared interface in the source file (YES) or not (NO)

See Also:

- ["DBA_JAVA_INNERS"](#) on page 3-83
- ["USER_JAVA_INNERS"](#) on page 3-167

ALL_JAVA_LAYOUTS

ALL_JAVA_LAYOUTS displays class layout information about the stored Java classes accessible to the current user.

Related Views

- [DBA_JAVA_LAYOUTS](#) displays class layout information about all stored Java classes in the database.
- [USER_JAVA_LAYOUTS](#) displays class layout information about the stored Java classes owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INTERFACES	NUMBER		Number of interfaces that this Java class implements
INNER_CLASSES	NUMBER		Number of inner classes that this Java class contains
FIELDS	NUMBER		Number of locally declared fields that this Java class contains
STATIC_FIELDS	NUMBER		Number of locally declared static fields that this Java class contains
METHODS	NUMBER		Number of locally declared methods that this Java class contains
STATIC_METHODS	NUMBER		Number of locally declared static methods that this Java class contains
NATIVE_METHODS	NUMBER		Number of locally declared native methods that this Java class contains

See Also:

- ["DBA_JAVA_LAYOUTS"](#) on page 3-83
- ["USER_JAVA_LAYOUTS"](#) on page 3-167

ALL_JAVA_METHODS

`ALL_JAVA_METHODS` displays method information about the stored Java classes accessible to the current user.

Related Views

- [DBA_JAVA_METHODS](#) displays method information about all stored Java classes in the database.
- [USER_JAVA_METHODS](#) displays method information about the stored Java classes owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the method
METHOD_NAME	VARCHAR2(4000)		Name of the method identified by the <code>METHOD_INDEX</code> column
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the method: <ul style="list-style-type: none"> ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the method is a static method (YES) or not (NO)

Column	Datatype	NULL	Description
IS_FINAL	VARCHAR2(3)		Indicates whether the method is a final method (YES) or not (NO)
IS_SYNCHRONIZED	VARCHAR2(3)		Indicates whether the method is a synchronized method (YES) or not (NO)
IS_NATIVE	VARCHAR2(3)		Indicates whether the method is a native method (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether the method is an abstract method (YES) or not (NO)
IS_STRICT	VARCHAR2(3)		Indicates whether the method is a strict method (YES) or not (NO)
ARGUMENTS	NUMBER		Number of arguments of the method
THROWS	NUMBER		Number of exceptions thrown by the method
ARRAY_DEPTH	NUMBER		Array depth of the return type of the method
BASE_TYPE	VARCHAR2(7)		Base type of the return type of the method: <ul style="list-style-type: none"> ▪ int ▪ long ▪ float ▪ double ▪ boolean ▪ byte ▪ char ▪ short ▪ class ▪ void
RETURN_CLASS	VARCHAR2(4000)		Actual class name of the return value if the base type is class

See Also:

- ["DBA_JAVA_METHODS"](#) on page 3-84
- ["USER_JAVA_METHODS"](#) on page 3-167

ALL_JAVA_NCOMPS

ALL_JAVA_NCOMPS displays ncomp-related information about the Java classes accessible to the current user.

Related Views

- DBA_JAVA_NCOMPS displays ncomp-related information about all Java classes in the database.
- USER_JAVA_NCOMPS displays ncomp-related information about the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class object
NAME	VARCHAR2(4000)		Name of the Java class object
SOURCE	VARCHAR2(4000)		ncomp source shown in this row
INITIALIZER	VARCHAR2(4000)		ncomp initializer shown in this row
LIBRARYFILE	VARCHAR2(4000)		ncomp library file shown in this row

Column	Datatype	NULL	Description
LIBRARY	VARCHAR2(4000)		ncomp library shown in this row

See Also:

- ["DBA_JAVA_NCOMPS"](#) on page 3-84
- ["USER_JAVA_NCOMPS"](#) on page 3-168

ALL_JAVA_RESOLVERS

ALL_JAVA_RESOLVERS displays information about resolvers of the Java classes accessible to the current user.

Related Views

- DBA_JAVA_RESOLVERS displays information about resolvers of all Java classes in the database.
- USER_JAVA_RESOLVERS displays information about resolvers of the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class object
NAME	VARCHAR2(4000)		Name of the Java class object
TERM_INDEX	NUMBER		Index of the resolver term in this row
PATTERN	VARCHAR2(4000)		Resolver pattern of the resolver term identified by the TERM_INDEX column
SCHEMA	VARCHAR2(64)		Resolver schema of the resolver term identified by the TERM_INDEX column

See Also:

- ["DBA_JAVA_RESOLVERS"](#) on page 3-84
- ["USER_JAVA_RESOLVERS"](#) on page 3-168

ALL_JAVA_THROWS

ALL_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes accessible to the current user.

Related Views

- DBA_JAVA_THROWS displays information about exceptions thrown from methods of all Java classes in the database.
- USER_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the throwing method of the exception

Column	Datatype	NULL	Description
METHOD_NAME	VARCHAR2 (4000)		Name of the throwing method of the exception
EXCEPTION_INDEX	NUMBER		Index of the exception
EXCEPTION_CLASS	VARCHAR2 (4000)		Class of the exception

See Also:

- ["DBA_JAVA_THROWS"](#) on page 3-84
- ["USER_JAVA_THROWS"](#) on page 3-168

ALL_JOBS

ALL_JOBS describes all jobs in the database that are accessible to the current user.

See Also: *Oracle Database Administrator's Guide*

Related Views

- DBA_JOBS describes all jobs in the database.
- USER_JOBS describes all jobs owned by the current user.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change this value.
LOG_USER	VARCHAR2 (30)	NOT NULL	Login user when the job was submitted
PRIV_USER	VARCHAR2 (30)	NOT NULL	User whose default privileges apply to this job
SCHEMA_USER	VARCHAR2 (30)	NOT NULL	Default schema used to parse the job For example, if the SCHEMA_USER is SCOTT and you submit the procedure HIRE_EMP as a job, the Oracle Database looks for SCOTT.HIRE_EMP
LAST_DATE	DATE		Date on which this job last successfully executed
LAST_SEC	VARCHAR2 (8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing (usually null if not executing)
THIS_SEC	VARCHAR2 (8)		Same as THIS_DATE. This is when the last successful execution started.
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2 (8)		Same as NEXT_DATE. This is when the last successful execution started.
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job, in seconds
BROKEN	VARCHAR2 (1)		Y: no attempt is made to run this job N: an attempt is made to run this job
INTERVAL	VARCHAR2 (200)	NOT NULL	A date function, evaluated at the start of execution, becomes next NEXT_DATE
FAILURES	NUMBER		Number of times the job has started and failed since its last success
WHAT	VARCHAR2 (4000)		Body of the anonymous PL/SQL block that the job executes
NLS_ENV	VARCHAR2 (4000)		Session parameters describing the NLS environment of the job

Column	Datatype	NULL	Description
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		ID of the instance that can execute or is executing the job. The default is 0.

ALL_JOIN_IND_COLUMNS

ALL_JOIN_IND_COLUMNS describes the join conditions of bitmap join indexes to which you have access. Bitmap join indexes are indexes built on a child table with an index key containing columns from associated parent tables, where all of the tables are connected through join conditions. There is one row for each join condition.

Related Views

- DBA_JOIN_IND_COLUMNS describes all join conditions existing in the database.
- USER_JOIN_IND_COLUMNS describes join conditions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	Description
INDEX_OWNER	VARCHAR2(30)	Bitmap join index owner
INDEX_NAME	VARCHAR2(30)	Name of bitmap join index
INNER_TABLE_OWNER	VARCHAR2(30)	Fact table owner
INNER_TABLE_NAME	VARCHAR2(30)	Name of the fact table
OUTER_TABLE_OWNER	VARCHAR2(30)	Dimension table owner
OUTER_TABLE_NAME	VARCHAR2(30)	Name of the dimension table
COL_NAME_INNER	VARCHAR2(4000)	Name of the fact table join column
COL_NAME_OUTER	VARCHAR2(4000)	Name of the dimension table join column

ALL_LIBRARIES

ALL_LIBRARIES describes all the libraries that are accessible to the current user.

Related Views

- DBA_LIBRARIES describes all libraries in the database.
- USER_LIBRARIES describes libraries owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the library
LIBRARY_NAME	VARCHAR2(30)	NOT NULL	Library name
FILE_SPEC	VARCHAR2(2000)		Operating system file specification associated with the library
DYNAMIC	VARCHAR2(1)		Whether the library is dynamically loadable (YES NO)
STATUS	VARCHAR2(7)		Status of the library (VALID, INVALID, or N/A)

ALL_LOB_PARTITIONS

ALL_LOB_PARTITIONS displays LOB partitions contained in tables accessible to the current user.

Related Views

- `DBA_LOB_PARTITIONS` describes all LOB partitions in the database.
- `USER_LOB_PARTITIONS` describes the LOB partitions owned by the current user. This view does not display the `TABLE_OWNER` column.

Column	Datatype	NULL	Description
<code>TABLE_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the table
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table
<code>COLUMN_NAME</code>	<code>VARCHAR2(4000)</code>		Name of the LOB column
<code>LOB_NAME</code>	<code>VARCHAR2(30)</code>		Name of the partitioned LOB item
<code>PARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table partition
<code>LOB_PARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Name of the LOB data partition
<code>LOB_INDPART_NAME</code>	<code>VARCHAR2(30)</code>		Name of the corresponding LOB index partition
<code>PARTITION_POSITION</code>	<code>NUMBER</code>		Position of the LOB data partition within the LOB item
<code>COMPOSITE</code>	<code>VARCHAR2(3)</code>		Whether the partition is composite (YES NO)
<code>CHUNK</code>	<code>NUMBER</code>		Value of the <code>CHUNK</code> attribute of the LOB data partition
<code>PCTVERSION</code>	<code>NUMBER</code>		Value of the <code>PCTVERSION</code> attribute of the LOB data partition
<code>CACHE</code>	<code>VARCHAR2(10)</code>		Whether and how the cluster is to be cached in the buffer cache (<code>CACHE</code> , <code>NOCACHE</code> , <code>CACHEREADS</code>)
<code>IN_ROW</code>	<code>VARCHAR2(3)</code>		Whether the <code>STORAGE IN ROW</code> attribute is enabled for the LOB data partition
<code>TABLESPACE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the tablespace containing the LOB data partition
<code>INITIAL_EXTENT</code>	<code>VARCHAR2(40)</code>		Size in bytes of the initial extent of the LOB data partition
<code>NEXT_EXTENT</code>	<code>VARCHAR2(40)</code>		Size in bytes of secondary extents of the LOB data partition
<code>MIN_EXTENTS</code>	<code>VARCHAR2(40)</code>		Minimum number of extents allowed in the segment of the LOB data partition
<code>MAX_EXTENTS</code>	<code>VARCHAR2(40)</code>		Maximum number of extents allowed in the segment of the LOB data partition
<code>PCT_INCREASE</code>	<code>VARCHAR2(40)</code>		Percentage increase in extent size for the LOB data partition
<code>FREELISTS</code>	<code>VARCHAR2(40)</code>		Number of process freelists allocated in the segment of the LOB data partition
<code>FREELIST_GROUPS</code>	<code>VARCHAR2(40)</code>		Number of freelist groups allocated in the segment of the LOB data partition
<code>LOGGING</code>	<code>VARCHAR2(7)</code>		Logging attribute of the LOB data partition
<code>BUFFER_POOL</code>	<code>VARCHAR2(7)</code>		Default buffer pool for the LOB partition blocks

ALL_LOB_SUBPARTITIONS

`ALL_LOB_SUBPARTITIONS` displays partition-level attributes of LOB data subpartitions accessible to the current user.

Related Views

- `DBA_LOB_SUBPARTITIONS` describes all LOB subpartitions in the database.
- `USER_LOB_SUBPARTITIONS` describes the LOB subpartitions owned by the current user. This view does not display the `TABLE_OWNER` column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB item
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition to which this LOB data subpartition belongs
SUBPARTITION_NAME	VARCHAR2(30)		Name of the table subpartition to which this LOB subpartition corresponds
LOB_SUBPARTITION_NAME	VARCHAR2(30)		Name of the LOB subpartition
LOB_INDSUBPART_NAME	VARCHAR2(30)		Name of corresponding LOB index subpartition
SUBPARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER	NOT NULL	Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(10)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)
IN_ROW	VARCHAR2(3)		Whether the STORAGE IN ROW attribute of the LOB data partition is enabled
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	NUMBER		Size in bytes of the initial extent for the LOB data partition
NEXT_EXTENT	NUMBER		Size in bytes of secondary extents for the LOB data partition
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	NUMBER		Percentage increase in extent size for the LOB data partition
FREELISTS	NUMBER		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(3)		Logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for the LOB data partition blocks

ALL_LOB_TEMPLATES

ALL_LOB_TEMPLATES describes the LOB subpartition templates accessible to the current user.

Related Views

- DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.
- USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)	NOT NULL	Owner of the table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
LOB_COL_NAME	VARCHAR2(4000)		Name of the LOB column
SUBPARTITION_NAME	VARCHAR2(34)	NOT NULL	Name of the subpartition
LOB_SEGMENT_NAME	VARCHAR2(34)	NOT NULL	Name of the LOB segment
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name of the subpartition

See Also:

- ["DBA_LOB_TEMPLATES"](#) on page 3-86
- ["USER_LOB_TEMPLATES"](#) on page 3-169

ALL_LOBS

ALL_LOBS displays the large objects (LOBs) contained in tables accessible to the current user. LOBs include binary large objects (BLOBs) and character large objects (CLOBs). Binary files (BFILEs) are stored outside the database, so they are not displayed by this view or the related views.

Related Views

- DBA_LOBS describes all LOBs in the database.
- USER_LOBS describes the LOBs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object containing the LOB
TABLE_NAME	VARCHAR2(30)		Name of the object containing the LOB
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)		Name of the LOB segment
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB segment
INDEX_NAME	VARCHAR2(30)		Name of the LOB index
CHUNK	NUMBER		Size (in bytes) of the LOB chunk as a unit of allocation or manipulation
PCTVERSION	NUMBER		Maximum percentage of the LOB space used for versioning
RETENTION	NUMBER		Maximum time duration for versioning of the LOB space
FREEPOOLS	NUMBER		Number of freepools for this LOB segment
CACHE	VARCHAR2(10)		Indicates whether and how the cluster is to be cached in the buffer cache (YES, NO, CACHEREADS)
LOGGING	VARCHAR2(7)		Indicates whether changes to the LOB are logged (YES) or not (NO)
IN_ROW	VARCHAR2(3)		Indicates whether some of the LOBs are stored inline with the base row (YES) or not (NO)
FORMAT	VARCHAR2(15)		Indicates whether the LOB storage format is dependent on the endianness of the platform: <ul style="list-style-type: none"> ■ NOT APPLICABLE ■ ENDIAN SPECIFIC ■ ENDIAN NEUTRAL

Column	Datatype	NULL	Description
PARTITIONED	VARCHAR2(3)		Indicates whether the LOB column is in a partitioned table (YES) or not (NO)

ALL_LOG_GROUP_COLUMNS

ALL_LOG_GROUP_COLUMNS describes columns that are accessible to the current user and that are specified in log groups.

Related Views

- DBA_LOG_GROUP_COLUMNS describes all columns in the database that are specified in log groups.
- USER_LOG_COLUMN_GROUPS describes columns that are owned by the current user and that are specified in log groups.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the log group definition
LOG_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the log group definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table in which the log group is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column or attribute of the object type column specified in the log group definition
POSITION	NUMBER		Original position of the column or attribute in the definition of the object
LOGGING_PROPERTY	VARCHAR2(6)		Indicates whether the column or attribute would be supplementally logged (LOG) or not (NO LOG)

ALL_LOG_GROUPS

ALL_LOG_GROUPS describes the log group definitions on the tables accessible to the current user.

Related Views

- DBA_LOG_GROUPS describes the log group definitions on all tables in the database.
- USER_LOG_GROUPS describes the log group definitions on the tables owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the log group definition
LOG_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the log group definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table on which the log group is defined
LOG_GROUP_TYPE	VARCHAR2(19)		Type of the log group: <ul style="list-style-type: none"> ■ PRIMARY KEY LOGGING ■ UNIQUE KEY LOGGING ■ FOREIGN KEY LOGGING ■ ALL COLUMN LOGGING ■ USER LOG GROUP
ALWAYS	VARCHAR2(11)		Y indicates the log group is logged any time a row is updated; N indicates the log group is logged any time a member column is updated.

Column	Datatype	NULL	Description
GENERATED	VARCHAR2 (14)		Indicates whether the name of the supplemental log group was system generated (GENERATED_NAME) or not (USER_NAME)

ALL_METHOD_PARAMS

ALL_METHOD_PARAMS describes the method parameters of the object types accessible to the current user.

Related Views

- DBA_METHOD_PARAMS describes the method parameters of all object types in the database.
- USER_METHOD_PARAMS describes the method parameters of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2 (30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2 (30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
PARAM_NAME	VARCHAR2 (30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number (position)
PARAM_MODE	VARCHAR2 (6)		Mode of the parameter (IN, OUT, IN OUT)
PARAM_TYPE_MOD	VARCHAR2 (7)		Whether this parameter is a REF to another object
PARAM_TYPE_OWNER	VARCHAR2 (30)		Owner of the type of the parameter
PARAM_TYPE_NAME	VARCHAR2 (30)		Name of the type of the parameter
CHARACTER_SET_NAME	VARCHAR2 (44)		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

See Also:

- ["DBA_METHOD_PARAMS"](#) on page 3-93
- ["USER_METHOD_PARAMS"](#) on page 3-169

ALL_METHOD_RESULTS

ALL_METHOD_RESULTS describes the method results of the object types accessible to the current user.

Related Views

- DBA_METHOD_RESULTS describes the method results of all object types in the database.
- USER_METHOD_RESULTS describes the method results of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
RESULT_TYPE_MOD	VARCHAR2(7)	NOT NULL	Whether this result is a REF to another object
RESULT_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the result
RESULT_TYPE_NAME	VARCHAR2(30)		Name of the type of the result
CHARACTER_SET_NAME	VARCHAR2(44)		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

See Also:

- ["DBA_METHOD_RESULTS"](#) on page 3-93
- ["USER_METHOD_RESULTS"](#) on page 3-169

ALL_MVIEW_AGGREGATES

ALL_MVIEW_AGGREGATES describes the grouping functions (aggregate operations) that appear in the SELECT list of materialized aggregate views accessible to the current user.

Related Views

- DBA_MVIEW_AGGREGATES describes all such grouping functions defined for all materialized views in the database.
- USER_MVIEW_AGGREGATES describes all such grouping functions defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that include references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8i and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this aggregation within the SELECT list. For the position of nonaggregate elements of the select list, see "ALL_MVIEW_KEYS" on page 2-63.
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Whether this aggregation is distinct (Y N)
MEASURE	LONG		SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*).

ALL_MVIEW_ANALYSIS

ALL_MVIEW_ANALYSIS describes the materialized views accessible to the current user. It provides additional information for analysis by applications. Minimal information is displayed for materialized views that do not support query rewrite (such as materialized views with remote master tables or nondeterministic functions).

Related Views

- DBA_MVIEW_ANALYSIS describes all such materialized views in the database.
- USER_MVIEW_ANALYSIS describes all such materialized views owned by the current user.

Note: All of the information in these views is also displayed in ALL_MVIEWS and its related views. Oracle recommends that you refer to ALL_MVIEWS for this information instead of these views.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the container table (see next column)
CONTAINER_NAME	VARCHAR2(30)		Name of the internal container in which the materialized view data is held. Normally this is the same as MVIEW_NAME. For materialized views created prior to Oracle8i, the Oracle Database attaches the 6-byte prefix SNAP\$. If MVIEW_NAME has more than 19 bytes, the Oracle Database truncates the name to 19 bytes and adds a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
LAST_REFRESH_SCN	NUMBER		System change number (SCN) of the last refresh operation
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		Default refresh method: FORCE, FAST, COMPLETE, or NONE
SUMMARY	VARCHAR2(1)		Whether this materialized view includes a GROUP BY clause or aggregation (Y N)
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds, for full refresh (defined only when SUMMARY = Y)
INCRFRESHTIM	NUMBER		Approximate refresh time, in seconds, for fast refresh (defined only when SUMMARY = Y)
CONTAINS_VIEWS	VARCHAR2(1)		Whether this materialized view contains a view in its definition (Y N)
UNUSABLE	VARCHAR2(1)		Whether this materialized view is UNUSABLE (inconsistent data) (Y N). A materialized view can be UNUSABLE if a system failure occurs during a full refresh.
RESTRICTED_SYNTAX	VARCHAR2(1)		Whether this materialized view had a restriction in its defining query that limits the use of query rewrite (Y N). More complete information is provided by the REWRITE_CAPABILITY column of the ALL_, DBA_, and USER_MVIEWS views.
INC_REFRESHABLE	VARCHAR2(1)		Whether this materialized view can be fast refreshed (Y N)

Column	Datatype	NULL	Description
KNOWN_STALE	VARCHAR2(1)		Whether the data contained in the materialized view is known to be inconsistent with the master table data because that has been updated since the last successful refresh (Y N)
INVALID	VARCHAR2(1)		Whether this materialized view is in an invalid state (inconsistent metadata) (Y N)
REWRITE_ENABLED	VARCHAR2(1)		Whether this materialized view is currently enabled for query rewrite (Y N)
QUERY_LEN	NUMBER		Length (in bytes) of the query field
QUERY	LONG		SELECT expression of the materialized view definition
REVISION	NUMBER		Reserved for internal use

ALL_MVIEW_COMMENTS

ALL_MVIEW_COMMENTS displays comments on the materialized views accessible to the current user.

Related Views

- DBA_MVIEW_COMMENTS displays comments on the materialized views in the database.
- USER_MVIEW_COMMENTS displays comments on the materialized views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
COMMENTS	VARCHAR2(4000)		Comment on the materialized view

See Also:

- ["DBA_MVIEW_COMMENTS"](#) on page 3-94
- ["USER_MVIEW_COMMENTS"](#) on page 3-170

ALL_MVIEW_DETAIL_RELATIONS

ALL_MVIEW_DETAIL_RELATIONS describes the named detail relations that are either specified in the FROM list of the subquery that defines a materialized view accessible to the current user, or that are indirectly referenced through views in that FROM list. Inline views in the materialized view definition are not represented in this view or the related views.

Related Views

- DBA_MVIEW_DETAIL_RELATIONS describes all such detail relations defined for all materialized views in the database.
- USER_MVIEW_DETAIL_RELATIONS describes such detail relations defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also

exclude materialized views that were created as *snapshots* prior to Oracle8i and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (that is, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, SNAPSHOT, CONTAINER, or UNDEFINED
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

ALL_MVIEW_JOINS

ALL_MVIEW_JOINS describes joins between two columns in the WHERE clause of the subquery that defines a materialized view accessible to the current user.

Related Views

- DBA_MVIEW_JOINS describes all such joins for all materialized views in the database.
- USER_MVIEW_JOINS describes such joins for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8i and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
DETAILOBJ1_OWNER	VARCHAR2(30)	NOT NULL	Owner of the first object in the join ¹
DETAILOBJ1_RELATION	VARCHAR2(30)	NOT NULL	Name of the first object in the join ¹
DETAILOBJ1_COLUMN	VARCHAR2(30)	NOT NULL	Join column of the first object in the join ¹
OPERATOR	CHAR(1)		Join operator ¹
OPERATOR_TYPE	VARCHAR2(1)		Whether the join is an inner or outer join ¹
DETAILOBJ2_OWNER	VARCHAR2(30)	NOT NULL	Owner of the second object in the join ¹
DETAILOBJ2_RELATION	VARCHAR2(30)	NOT NULL	Name of the second object in the join ¹
DETAILOBJ2_COLUMN	VARCHAR2(30)	NOT NULL	Join column of the second object in the join ¹

¹ These rows relate only to materialized join views and materialized aggregate views. They describe the two detail objects of a materialized view join.

ALL_MVIEW_KEYS

ALL_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views accessible to the current user are based.

Related Views

- **DBA_MVIEW_KEYS** describes such columns and expressions for all materialized views in the database.
- **USER_MVIEW_KEYS** describes such columns and expressions for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as `SYSDATE` or `USER`. These views also exclude materialized views that were created as *snapshots* prior to Oracle8i and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of the column in the container table
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (for example, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(5)		Detail object type (VIEW TABLE)
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

ALL_MVIEW_LOGS

ALL_MVIEW_LOGS describes all materialized view logs accessible to the current user.

Related Views

- **DBA_MVIEW_LOGS** describes all materialized view logs in the database.
- **USER_MVIEW_LOGS** describes all materialized view logs owned by the current user.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)		Owner of the materialized view log
MASTER	VARCHAR2(30)		Name of the master table or master materialized view whose changes are logged
LOG_TABLE	VARCHAR2(30)		Name of the table where the changes to the master table or master materialized view are logged
LOG_TRIGGER	VARCHAR2(30)		Obsolete with the release of Oracle8i and higher. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserted rows into the log.
ROWIDS	VARCHAR2(3)		If YES, records rowid information
PRIMARY_KEY	VARCHAR2(3)		If YES, records primary key information
OBJECT_ID	VARCHAR2(3)		If YES, records object identifier information in an object table
FILTER_COLUMNS	VARCHAR2(3)		If YES, records filter columns
SEQUENCE	VARCHAR2(3)		If YES, records the sequence value, which provides additional ordering information
INCLUDE_NEW_VALUES	VARCHAR2(3)		If YES, records both old and new values. If NO, records old values, but does not record new values.

See Also:

- ["DBA_MVIEW_LOGS"](#) on page 3-94
- ["USER_MVIEW_LOGS"](#) on page 3-170

ALL_MVIEW_REFRESH_TIMES

ALL_MVIEW_REFRESH_TIMES describes refresh times of the materialized views accessible to the current user.

Related Views

- DBA_MVIEW_REFRESH_TIMES describes refresh times of all materialized views in the database.
- USER_MVIEW_REFRESH_TIMES describes refresh times of the materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

See Also:

- ["DBA_MVIEW_REFRESH_TIMES"](#) on page 3-95
- ["USER_MVIEW_REFRESH_TIMES"](#) on page 3-170

ALL_MVIEWS

ALL_MVIEWS describes all materialized views accessible to the current user.

Related views

- DBA_MVIEWS describes all materialized views in the database.
- USER_MVIEWS describes all materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Schema in which the materialized view was created
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
CONTAINER_NAME	VARCHAR2(30)	NOT NULL	Name of the container in which the materialized view's data is held. Normally this is the same as MVIEW_NAME. For materialized views created prior to Oracle8i, the Oracle Database attaches the 6-byte prefix SNAP\$_. If MVIEW_NAME has more than 19 bytes, then the Oracle Database truncates the name to 19 bytes and may add a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
QUERY	LONG		Query that defines the materialized view
QUERY_LEN	NUMBER(38)		Length (in bytes) of the defining query
UPDATABLE	VARCHAR2(1)		Indicates whether the materialized view is updatable (Y) or not (N)

Column	Datatype	NULL	Description
UPDATE_LOG	VARCHAR2(30)		For updatable materialized views, the filename of the update log
MASTER_ROLLBACK_SEG	VARCHAR2(30)		Rollback segment for the master site or the master materialized view site
MASTER_LINK	VARCHAR2(128)		Database link for the master site or the master materialized view site
REWRITE_ENABLED	VARCHAR2(1)		Indicates whether rewrite is enabled (Y) or not (N)
REWRITE_CAPABILITY	VARCHAR2(9)		Indicates whether the materialized view is eligible for rewrite, and if so, what rules must be followed: <ul style="list-style-type: none"> ■ NONE - Materialized view cannot be used for rewrite, because rewrite is disallowed or prevented ■ TEXTMATCH - Defining query of the materialized view contained restrictions on the use of query rewrite ■ GENERAL - Defining query of the materialized view contained no restrictions on the use of query rewrite, so the Oracle Database can apply any rewrite rule that is supported
REFRESH_MODE	VARCHAR2(6)		Refresh mode of the materialized view: <ul style="list-style-type: none"> ■ DEMAND - Oracle Database refreshes this materialized view whenever an appropriate refresh procedure is called ■ COMMIT - Oracle Database refreshes this materialized view when a transaction on one of the materialized view's masters commits ■ NEVER - Oracle Database never refreshes this materialized view
REFRESH_METHOD	VARCHAR2(8)		Default method used to refresh the materialized view (can be overridden through the API): <ul style="list-style-type: none"> ■ COMPLETE (C) - Materialized view is completely refreshed from the masters ■ FORCE (?) - Oracle Database performs a fast refresh if possible, otherwise a complete refresh ■ FAST (F) - Oracle Database performs an incremental refresh applying changes that correspond to changes in the masters since the last refresh ■ NEVER (N) - User specified that the Oracle Database should not refresh this materialized view
BUILD_MODE	VARCHAR2(9)		Indicates how the materialized view was populated during creation: <ul style="list-style-type: none"> ■ IMMEDIATE - Populated from the masters during creation ■ DEFERRED - Not populated during creation. Must be explicitly populated later by the user. ■ PREBUILT - Populated with an existing table during creation. The relationship of the contents of this prebuilt table to the materialized view's masters is unknown to the Oracle Database.

Column	Datatype	NULL	Description
FAST_REFRESHABLE	VARCHAR2 (18)		Indicates whether the materialized view is eligible for incremental (fast) refresh. The Oracle Database calculates this value statically, based on the materialized view definition query: <ul style="list-style-type: none"> NO - Materialized view is not fast refreshable, and hence is complex DIRLOAD - Fast refresh is supported only for direct loads DML - Fast refresh is supported only for DML operations DIRLOAD_DML - Fast refresh is supported for both direct loads and DML operations DIRLOAD_LIMITEDDML - Fast refresh is supported for direct loads and a subset of DML operations
LAST_REFRESH_TYPE	VARCHAR2 (8)		Method used for the most recent refresh: <ul style="list-style-type: none"> COMPLETE - Most recent refresh was complete FAST - Most recent refresh was fast (incremental) NA - Materialized view has not yet been refreshed (for example, if it was created DEFERRED)
LAST_REFRESH_DATE	DATE		Date on which the materialized view was most recently refreshed. Blank if not yet populated.
STALENESS	VARCHAR2 (19)		Relationship between the contents of the materialized view and the contents of the materialized view's masters: <ul style="list-style-type: none"> FRESH - Materialized view is a read-consistent view of the current state of its masters STALE - Materialized view is out of date because one or more of its masters has changed. If the materialized view was FRESH before it became STALE, then it is a read-consistent view of a former state of its masters. UNUSABLE - Materialized view is not a read-consistent view of its masters from any point in time UNKNOWN - Oracle Database does not know whether the materialized view is in a read-consistent view of its masters from any point in time (this is the case for materialized views created on prebuilt tables) UNDEFINED - Materialized view has remote masters. The concept of staleness is not defined for such materialized views.
AFTER_FAST_REFRESH	VARCHAR2 (19)		Specifies the staleness value that will occur if a fast refresh is applied to this materialized view. Its values are the same as for the STALENESS column, plus the value NA, which is used when fast refresh is not applicable to this materialized view.
UNKNOWN_PREBUILT	VARCHAR2 (1)		Indicates whether the materialized view is prebuilt (Y) or not (N)
UNKNOWN_PLSQL_FUNC	VARCHAR2 (1)		Indicates whether the materialized view contains PL/SQL functions (Y) or not (N)
UNKNOWN_EXTERNAL_TABLE	VARCHAR2 (1)		Indicates whether the materialized view contains external tables (Y) or not (N)
UNKNOWN_CONSIDER_FRESH	VARCHAR2 (1)		Indicates whether the materialized view is considered fresh (Y) or not (N)
UNKNOWN_IMPORT	VARCHAR2 (1)		Indicates whether the materialized view is imported (Y) or not (N)
UNKNOWN_TRUSTED_FD	VARCHAR2 (1)		Indicates whether the materialized view uses trusted constraints for refresh (Y) or not (N)

Column	Datatype	NULL	Description
COMPILE_STATE	VARCHAR2(19)		Validity of the materialized view with respect to the objects upon which it depends: <ul style="list-style-type: none"> VALID - Materialized view has been validated without error, and no object upon which it depends has changed since the last validation NEEDS_COMPILE - Some object upon which the materialized view depends has changed (other than normal DML changes). An ALTER MATERIALIZED VIEW...COMPILE statement is required to validate this materialized view. ERROR - Materialized view has been validated with one or more errors
USE_NO_INDEX	VARCHAR2(1)		Indicates whether the materialized view was created using the USING NO INDEX clause (Y) or the materialized view was created with the default index (N). The USING NO INDEX clause suppresses the creation of the default index.
STALE_SINCE	DATE		Time from when the materialized view became stale

See Also:

- ["DBA_MVIEWS"](#) on page 3-95
- ["USER_MVIEWS"](#) on page 3-171
- *Oracle Database Advanced Replication* for more information on materialized views to support replication
- *Oracle Data Warehousing Guide* for more information on materialized views to support data warehousing

ALL_NESTED_TABLE_COLS

ALL_NESTED_TABLE_COLS describes the columns of the nested tables accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

Related Views

- DBA_NESTED_TABLE_COLS describes the columns of all nested tables in the database.
- USER_NESTED_TABLE_COLS describes the columns of the nested tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the nested table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the nested table
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes

Column	Datatype	NULL	Description
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2 (1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER		Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column ¹
LOW_VALUE	RAW (3 2)		Low value in the column ¹
HIGH_VALUE	RAW (3 2)		High value in the column ¹
DENSITY	NUMBER		Density of the column ¹
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column Note: The number of buckets in a histogram is specified in the SIZE parameter of the SQL statement ANALYZE. However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2 (4 4)		Name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2 (3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2 (3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2 (1)		B C. B indicates that the column uses BYTE length semantics. C indicates that the column uses CHAR length semantics. NULL indicates the datatype is not any of the following: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2 (1)		Indicates whether the column data is in release 8.0 image format (Y) or not (N)
DATA_UPGRADED	VARCHAR2 (1)		Indicates whether the column data has been upgraded to the latest type version format (Y) or not (N)

Column	Datatype	NULL	Description
HIDDEN_COLUMN	VARCHAR2(3)		Indicates whether the column is a hidden column (YES) or not (NO)
VIRTUAL_COLUMN	VARCHAR2(3)		Indicates whether the column is a virtual column (YES) or not (NO)
SEGMENT_COLUMN_ID	NUMBER		Sequence number of the column in the segment
INTERNAL_COLUMN_ID	NUMBER	NOT NULL	Internal sequence number of the column
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: <ul style="list-style-type: none"> ▪ NONE ▪ FREQUENCY ▪ HEIGHT BALANCED
QUALIFIED_COL_NAME	VARCHAR2(4000)		Qualified column name

¹ These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views.

ALL_NESTED_TABLES

ALL_NESTED_TABLES describes the nested tables in tables accessible to the current user.

Related Views

- DBA_NESTED_TABLES describes all nested tables in the database.
- USER_NESTED_TABLES describes nested tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the nested table
TABLE_NAME	VARCHAR2(30)		Name of the nested table
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of which the nested table was created
TABLE_TYPE_NAME	VARCHAR2(30)		Name of the type of the nested table
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the parent table containing the nested table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Column name of the parent table that corresponds to the nested table
STORAGE_SPEC	VARCHAR2(30)		Whether storage for the nested table is USER-SPECIFIED or DEFAULT
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column (LOCATOR VALUE)
ELEMENT_SUBSTITUTABLE	VARCHAR2(25)		Indicates whether the nested table element is substitutable (Y) or not (N)

ALL_OBJ_COLATTRS

ALL_OBJ_COLATTRS describes object columns and attributes contained in the tables accessible to the current user.

Related Views

- DBA_OBJ_COLATTRS describes object columns and attributes contained in all tables in the database.
- USER_OBJ_COLATTRS describes object columns and attributes contained in the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table containing the object column or attribute
COLUMN_NAME	VARCHAR2(4000)		Fully qualified name of the object column or attribute
SUBSTITUTABLE	VARCHAR2(15)		Indicates whether the column is substitutable (Y) or not (N)

See Also:

- ["DBA_OBJ_COLATTRS"](#) on page 3-96
- ["USER_OBJ_COLATTRS"](#) on page 3-171

ALL_OBJECT_TABLES

ALL_OBJECT_TABLES describes the object tables accessible to the current user.

Related Views

- DBA_OBJECT_TABLES describes all object tables in the database.
- USER_OBJECT_TABLES describes the object tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; null for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
PCT_FREE	NUMBER		Minimum percentage of free space in a block; null for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; null for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; null for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; null for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); null for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); null for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; null for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; null for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; null for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; null for partitioned tables

Column	Datatype	NULL	Description
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment; null for partitioned tables
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Whether the table has been backed up since last modification
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of chained rows in the table
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average free space of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of parallel execution processes per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)
TABLE_LOCK	VARCHAR2(8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Whether the table is partitioned (YES NO)
IOT_TYPE	VARCHAR2(12)		If index-organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		Whether the object ID (OID) is USER-DEFINED or SYSTEM-GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the table
TABLE_TYPE	VARCHAR2(30)		Type of the table
TEMPORARY	VARCHAR2(1)		Whether this is a temporary table
SECONDARY	VARCHAR2(1)		Whether the object table is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y N)
NESTED	VARCHAR2(3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for table blocks
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is ENABLED or DISABLED
GLOBAL_STATS	VARCHAR2(3)		For partitioned object tables, indicates whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

Column	Datatype	NULL	Description
DURATION	VARCHAR2 (15)		Indicates the duration of a temporary table: <ul style="list-style-type: none"> ■ SYS\$SESSION: the rows are preserved for the duration of the session ■ SYS\$TRANSACTION: the rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2 (8)		Whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2 (3)		Whether the table has the MONITORING attribute set
CLUSTER_OWNER	VARCHAR2 (30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2 (8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2 (8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); null for partitioned tables
DROPPED	VARCHAR2 (3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); null for partitioned tables

See Also:

- ["DBA_OBJECT_TABLES"](#) on page 3-97
- ["USER_OBJECT_TABLES"](#) on page 3-171

ALL_OBJECTS

ALL_OBJECTS describes all objects accessible to the current user.

Related Views

- DBA_OBJECTS describes all objects in the database.
- USER_OBJECTS describes all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2 (30)		Name of the subobject (for example, partition)
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object
DATA_OBJECT_ID	NUMBER		Dictionary object number of the segment that contains the object Note: OBJECT_ID and DATA_OBJECT_ID display data dictionary metadata. Do not confuse these numbers with the unique 16-byte object identifier (<i>object ID</i>) that the Oracle Database assigns to row objects in object tables in the system.
OBJECT_TYPE	VARCHAR2 (19)		Type of the object (such as TABLE, INDEX)
CREATED	DATE	NOT NULL	Timestamp for the creation of the object
LAST_DDL_TIME	DATE	NOT NULL	Timestamp for the last modification of the object resulting from a DDL statement (including grants and revokes)

Column	Datatype	NULL	Description
TIMESTAMP	VARCHAR2(20)		Timestamp for the specification of the object (character data)
STATUS	VARCHAR2(7)		Status of the object (VALID, INVALID, or N/A)
TEMPORARY	VARCHAR2(1)		Whether the object is temporary (the current session can see only data that it placed in this object itself)
GENERATED	VARCHAR2(1)		Was the name of this object system generated? (Y N)
SECONDARY	VARCHAR2(1)		Whether this is a secondary object created by the <code>ODCIIndexCreate</code> method of the Oracle Data Cartridge (Y N)

ALL_OPANCILLARY

ALL_OPANCILLARY describes operators whose bindings are ancillary to other (primary) operators.

Related Views

- DBA_OPANCILLARY provides such information about all operators in the database.
- USER_OPANCILLARY provides such information about operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number the of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of the primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of the primary operator

ALL_OPARGUMENTS

ALL_OPARGUMENTS describes arguments for each operator binding accessible to the current user.

Related Views

- DBA_OPARGUMENTS describes arguments of all operator bindings in the database.
- USER_OPARGUMENTS describes arguments of all operator bindings owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator argument
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator argument
BINDING#	NUMBER	NOT NULL	Binding number of the operator argument
POSITION	NUMBER	NOT NULL	Position of the operator argument (1, 2, 3, ...)
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

ALL_OPBINDINGS

ALL_OPBINDINGS describes the binding functions and methods on the operators accessible to the current user.

Related Views

- DBA_OPBINDINGS describes the binding functions and methods on all operators in the database.
- USER_OPBINDINGS describes the binding functions and methods on the operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		Name of the schema of the return type if the return type of the binding is an object type
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT or SCAN CONTEXT, then this column displays the schema of the implementation type used by the functional implementation of the operator as a scan context (null if the operator was created without this syntax). See Also: the CREATE OPERATOR statement in <i>Oracle Database SQL Reference</i>
IMPLEMENTATION_TYPE	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT or SCAN CONTEXT, then this column displays the name of the implementation type used by the functional implementation of the operator as a scan context (null if the operator was created without this syntax). See Also: the CREATE OPERATOR statement in <i>Oracle Database SQL Reference</i>
PROPERTY	VARCHAR2(43)		Property of the operator binding: <ul style="list-style-type: none"> ■ WITH INDEX CONTEXT ■ COMPUTE ANCILLARY DATA ■ ANCILLARY TO ■ WITH COLUMN CONTEXT ■ WITH INDEX, COLUMN CONTEXT ■ COMPUTE ANCILLARY DATA, WITH COLUMN CONTEXT

See Also:

- "DBA_OPBINDINGS" on page 3-97
- "USER_OPBINDINGS" on page 3-172

ALL_OPERATOR_COMMENTS

ALL_OPERATOR_COMMENTS displays comments for the user-defined operators accessible to the current user.

Related Views

- `DBA_OPERATOR_COMMENTS` displays comments for all user-defined operators in the database.
- `USER_OPERATOR_COMMENTS` displays comments for the user-defined operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the user-defined operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the user-defined operator
COMMENTS	VARCHAR2(4000)		Comment for the user-defined operator

See Also:

- ["DBA_OPERATOR_COMMENTS"](#) on page 3-97
- ["USER_OPERATOR_COMMENTS"](#) on page 3-172

ALL_OPERATORS

`ALL_OPERATORS` describes operators accessible to the current user.

Related Views

- `DBA_OPERATORS` describes all operators in the database.
- `USER_OPERATORS` describes all operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

See Also:

- ["DBA_OPERATORS"](#) on page 3-97
- ["USER_OPERATORS"](#) on page 3-172

ALL_OUTLINE_HINTS

`ALL_OUTLINE_HINTS` describes the set of hints stored in outlines accessible to the current user.

Related Views

- `DBA_OUTLINE_HINTS` describes such hints for all outlines in the database. This is the only one of the three views that displays the `OWNER` column.
- `USER_OUTLINE_HINTS` describes such hints for all outlines owned by the current user.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline
OWNER	VARCHAR2(30)		Name of the user who created the outline

Column	Datatype	NULL	Description
NODE	NUMBER		ID of the query or subquery to which the hint applies. The top-level query is labeled 1. Subqueries are assigned sequentially numbered labels, starting with 2.
STAGE	NUMBER		Outline hints can be applied at three different stages during the compilation process. This column indicates the stage at which this hint was applied.
JOIN_POS	NUMBER		Position of the table in the join order. The value is 0 for all hints except access method hints, which identify a table to which the hint and the join position apply.
HINT	VARCHAR2(512)		Text of the hint

ALL_OUTLINES

ALL_OUTLINES describes all stored outlines accessible to the current user.

Related Views

- DBA_OUTLINES describes all stored outlines in the database. This is the only one of the three views that displays the OWNER column.
- USER_OUTLINES describes all stored outlines owned by the current user.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		User-specified or generated name of the stored outline. The name must be of a form that can be expressed in SQL.
OWNER	VARCHAR2(30)		Name of the user who created the outline
CATEGORY	VARCHAR2(30)		User-defined name of the category to which the outline belongs
USED	VARCHAR2(6)		Indicates whether the outline has ever been used (USED) or not (UNUSED)
TIMESTAMP	DATE		Timestamp of outline creation
VERSION	VARCHAR2(64)		Oracle version that created the outline
SQL_TEXT	LONG		SQL text of the query, including any hints that were a part of the original statement. If bind variables are included, the variable names are stored as SQL text, not the values that are assigned to the variables. Note: This field may contain sensitive information about your database or application. Therefore, use discretion when granting SELECT or VIEW object privileges on these views.
SIGNATURE	RAW(16)		Signature uniquely identifying the outline SQL text
COMPATIBLE	VARCHAR2(12)		Indicates whether the outline hints were compatible across a migration (COMPATIBLE) or not (INCOMPATIBLE)
ENABLED	VARCHAR2(8)		Indicates whether the outline is enabled (ENABLED) or disabled (DISABLED)
FORMAT	VARCHAR2(6)		Hint format: <ul style="list-style-type: none"> ■ NORMAL ■ LOCAL

ALL_PART_COL_STATISTICS

ALL_PART_COL_STATISTICS provides column statistics and histogram information for table partitions accessible to the current user.

Related Views

- `DBA_PART_COL_STATISTICS` provides such information for all table partitions in the database.
- `USER_PART_COL_STATISTICS` provides such information for all partitions of tables owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the partitioned table
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the table
<code>PARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table partition
<code>COLUMN_NAME</code>	<code>VARCHAR2(4000)</code>		Name of the column
<code>NUM_DISTINCT</code>	<code>NUMBER</code>		Number of distinct values in the column
<code>LOW_VALUE</code>	<code>RAW(32)</code>		Low value in the column
<code>HIGH_VALUE</code>	<code>RAW(32)</code>		High value in the column
<code>DENSITY</code>	<code>NUMBER</code>		Density of the column
<code>NUM_NULLS</code>	<code>NUMBER</code>		Number of nulls in the column
<code>NUM_BUCKETS</code>	<code>NUMBER</code>		Number of buckets in histogram for the column
<code>SAMPLE_SIZE</code>	<code>NUMBER</code>		Sample size used in analyzing this column
<code>LAST_ANALYZED</code>	<code>DATE</code>		Date on which this column was most recently analyzed
<code>GLOBAL_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether column statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
<code>USER_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether statistics were entered directly by the user (YES) or not (NO)
<code>AVG_COL_LEN</code>	<code>NUMBER</code>		Average length of the column (in bytes)
<code>HISTOGRAM</code>	<code>VARCHAR2(15)</code>		Indicates existence/type of histogram: <ul style="list-style-type: none"> ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

ALL_PART_HISTOGRAMS

`ALL_PART_HISTOGRAMS` provides the histogram data (endpoints per histogram) for histograms on table partitions accessible to the current user.

Note: These views are populated only if you collect statistics on the index using the `ANALYZE` statement or the `DBMS_STATS` package.

Related Views

- `DBA_PART_HISTOGRAMS` provides such information for all table partitions in the database.
- `USER_PART_HISTOGRAMS` provides such information for all partitions of tables owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the table
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(30)		Name of the column
BUCKET_NUMBER	NUMBER		Bucket number of the histogram
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket

ALL_PART_INDEXES

ALL_PART_INDEXES displays the object-level partitioning information for the partitioned indexes accessible to the current user.

Related Views

- DBA_PART_INDEXES displays the object-level partitioning information for all partitioned indexes in the database.
- USER_PART_INDEXES displays the object-level partitioning information for the partitioned indexes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		Type of partitioning method: <ul style="list-style-type: none"> ■ RANGE ■ HASH ■ SYSTEM ■ LIST
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of composite partitioning method: <ul style="list-style-type: none"> ■ NONE ■ HASH ■ SYSTEM ■ LIST
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in the index
DEF_SUBPARTITION_COUNT	NUMBER		For a composite-partitioned index, the default number of subpartitions, if specified
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY_COUNT	NUMBER		For a composite-partitioned index, the number of columns in the subpartitioning key
LOCALITY	VARCHAR2(6)		Indicates whether the partitioned index is local (LOCAL) or global (GLOBAL)
ALIGNMENT	VARCHAR2(12)		Indicates whether the partitioned index is PREFIXED or NON_PREFIXED
DEF_TABLESPACE_NAME	VARCHAR2(30)		For a local index, the default tablespace to be used when adding or splitting a table partition
DEF_PCT_FREE	NUMBER	NOT NULL	For a local index, the default PCTFREE value to be used when adding a table partition

Column	Datatype	NULL	Description
DEF_INI_TRANS	NUMBER	NOT NULL	For a local index, the default INITRANS to be used when adding a table partition
DEF_MAX_TRANS	NUMBER	NOT NULL	For a local index, the default MAXTRANS to be used when adding a table partition
DEF_INITIAL_EXTENT	VARCHAR2(40)	NOT NULL	For a local index, the default INITIAL value to be used when adding a table partition, or DEFAULT if no INITIAL value was specified
DEF_NEXT_EXTENT	VARCHAR2(40)	NOT NULL	For a local index, the default NEXT, or DEFAULT if no NEXT value was specified
DEF_MIN_EXTENTS	VARCHAR2(40)	NOT NULL	For a local index, the default MINEXTENTS value to be used when adding a table partition, or DEFAULT if no MINEXTENTS value was specified
DEF_MAX_EXTENTS	VARCHAR2(40)	NOT NULL	For a local index, the default MAXEXTENTS value to be used when adding a table partition, or DEFAULT if no MAXEXTENTS value was specified
DEF_PCT_INCREASE	VARCHAR2(40)	NOT NULL	For a local index, the default PCTINCREASE value to be used when adding a table partition, or DEFAULT if no PCTINCREASE value was specified
DEF_FREELISTS	NUMBER	NOT NULL	For a local index, the default FREELISTS value to be used when adding a table partition, or DEFAULT if no FREELISTS value was specified
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	For a local index, the default FREELIST_GROUPS value to be used when adding a table partition, or DEFAULT if no FREELIST_GROUPS value was specified
DEF_LOGGING	VARCHAR2(7)		For a local index, the default LOGGING attribute to be used when adding a table partition, or DEFAULT if no LOGGING attribute was specified
DEF_BUFFER_POOL	VARCHAR2(7)		For a local index, the default buffer pool to be used when adding a table partition
DEF_PARAMETERS	VARCHAR2(1000)		Default parameter string for domain indexes

See Also:

- ["DBA_PART_INDEXES"](#) on page 3-99
- ["USER_PART_INDEXES"](#) on page 3-173

ALL_PART_KEY_COLUMNS

ALL_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects accessible to the current user.

Related Views

- DBA_PART_KEY_COLUMNS describes the partitioning key columns for all partitioned objects in the database.
- USER_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the partitioned table or index
NAME	VARCHAR2(30)		Name of the partitioned table or index

Column	Datatype	NULL	Description
OBJECT_TYPE	CHAR(5)		Object type: <ul style="list-style-type: none"> ■ TABLE ■ INDEX
COLUMN_NAME	VARCHAR2(30)		Name of the column
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

See Also:

- ["DBA_PART_KEY_COLUMNS"](#) on page 3-99
- ["USER_PART_KEY_COLUMNS"](#) on page 3-173

ALL_PART_LOBS

ALL_PART_LOBS provides table-level information about partitioned LOBs that are accessible to the current user, including default attributes for LOB data partitions.

Related Views

- DBA_PART_LOBS provides such information for all partitioned LOBs in the database.
- USER_PART_LOBS provides such information for all partitioned LOBs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table containing LOB(s)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table containing LOB(s)
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default value of CHUNK for a LOB data partition to be used when adding a partition
DEF_PCTVERSION	NUMBER	NOT NULL	Default value of PCTVERSION for a LOB data partition to be used when adding a partition
DEF_CACHE	VARCHAR2(10)		Whether and how the cluster is to be cached by default in the buffer cache (CACHE, NOCACHE, CACHEREADS)
DEF_IN_ROW	VARCHAR2(3)		Whether LOB data < 4000 bytes is stored in the row (inline)—that is, whether ENABLE STORAGE IN ROW was specified when the LOB column was created or last altered
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default tablespace for a LOB data partition to be used when adding a partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL for a LOB data partition to be used when adding a partition
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT for a LOB data partition to be used when adding a partition
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENT for a LOB data partition to be used when adding a partition
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS for a LOB data partition to be used when adding a partition
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE for a LOB data partition to be used when adding a partition

Column	Datatype	NULL	Description
DEF_FREELISTS	VARCHAR2(40)		Default value of FREELISTS for a LOB data partition to be used when adding a partition
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default value of FREELIST_GROUPS for a LOB data partition to be used when adding a partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition to be used when adding a partition
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for a LOB data partition to be used when adding a partition

ALL_PART_TABLES

ALL_PART_TABLES displays the object-level partitioning information for the partitioned tables accessible to the current user.

Related Views

- DBA_PART_TABLES displays the object-level partitioning information for all partitioned tables in the database.
- USER_PART_TABLES displays the object-level partitioning information for the partitioned tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the partitioned table
TABLE_NAME	VARCHAR2(30)		Name of the partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		Type of partitioning method: <ul style="list-style-type: none"> ■ RANGE ■ HASH ■ SYSTEM ■ LIST
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of composite partitioning method: <ul style="list-style-type: none"> ■ NONE ■ HASH ■ SYSTEM ■ LIST
PARTITION_COUNT	NUMBER		Number of partitions in the table
DEF_SUBPARTITION_COUNT	NUMBER		For a composite-partitioned table, the default number of subpartitions, if specified
PARTITIONING_KEY_COUNT	NUMBER		Number of columns in the partitioning key
SUBPARTITIONING_KEY_COUNT	NUMBER		For a composite-partitioned table, the number of columns in the subpartitioning key
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default tablespace to be used when adding a partition
DEF_PCT_FREE	NUMBER		Default value of PCTFREE to be used when adding a partition
DEF_PCT_USED	NUMBER		Default value of PCTUSED to be used when adding a partition
DEF_INI_TRANS	NUMBER		Default value of INITRANS to be used when adding a partition
DEF_MAX_TRANS	NUMBER		Default value of MAXTRANS to be used when adding a partition

Column	Datatype	NULL	Description
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL to be used when adding a partition. DEFAULT if INITIAL was not specified.
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT to be used when adding a partition. DEFAULT if NEXT was not specified.
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENTS to be used when adding a partition. DEFAULT if MINEXTENTS was not specified.
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS to be used when adding a partition. DEFAULT if MAXEXTENTS was not specified.
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE to be used when adding a partition. DEFAULT if PCTINCREASE was not specified.
DEF_FREELISTS	NUMBER		Default value of FREELISTS to be used when adding a partition
DEF_FREELIST_GROUPS	NUMBER		Default value of FREELIST_GROUPS to be used when adding a partition
DEF_LOGGING	VARCHAR2(7)		Default logging attribute to be used when adding a partition (NONE YES NO)
DEF_COMPRESSION	VARCHAR2(8)		Default compression to be used when adding a partition (NONE ENABLED DISABLED)
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used when adding a partition (DEFAULT KEEP RECYCLE)

See Also:

- ["DBA_PART_TABLES"](#) on page 3-99
- ["USER_PART_TABLES"](#) on page 3-173

ALL_PARTIAL_DROP_TABS

ALL_PARTIAL_DROP_TABS describes tables accessible to the current user that have partially completed DROP COLUMN operations. Such operations might have been interrupted by the user or by a system crash.

Related Views

- DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations.
- USER_PARTIAL_DROP_TABS describes tables in the schema of the current user that have partially completed DROP COLUMN operations. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

ALL_PENDING_CONV_TABLES

ALL_PENDING_CONV_TABLES describes the pending conversion tables (tables which are not upgraded to the latest type version) accessible to the current user.

Related Views

- [DBA_PENDING_CONV_TABLES](#) describes all pending conversion tables in the database.
- [USER_PENDING_CONV_TABLES](#) describes the pending conversion tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

See Also:

- ["DBA_PENDING_CONV_TABLES"](#) on page 3-100
- ["USER_PENDING_CONV_TABLES"](#) on page 3-174

ALL_PLSQL_OBJECT_SETTINGS

ALL_PLSQL_OBJECT_SETTINGS displays compiler settings for the stored objects accessible to the current user.

Related Views

- [DBA_PLSQL_OBJECT_SETTINGS](#) displays compiler settings for all stored objects in the database.
- [USER_PLSQL_OBJECT_SETTINGS](#) displays compiler settings for the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: <ul style="list-style-type: none"> ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY
PLSQL_OPTIMIZE_LEVEL	NUMBER		Optimization level that was used to compile the object
PLSQL_CODE_TYPE	VARCHAR2(4000)		Compilation mode for the object
PLSQL_DEBUG	VARCHAR2(4000)		Indicates whether or not the object was compiled for debugging
PLSQL_WARNINGS	VARCHAR2(4000)		Compiler warning settings that were used to compile the object
NLS_LENGTH_SEMANTICS	VARCHAR2(4000)		NLS length semantics that were used to compile the object

See Also:

- ["DBA_PLSQL_OBJECT_SETTINGS"](#) on page 3-100
- ["USER_PLSQL_OBJECT_SETTINGS"](#) on page 3-174

ALL_POLICIES

ALL_POLICIES describes the security policies on the synonyms, tables, and views accessible to the current user.

Related Views

- DBA_POLICIES describes all security policies in the database.
- USER_POLICIES describes the security policies on the synonyms, tables, and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
POLICY_GROUP	VARCHAR2(30)	NOT NULL	Name of the policy group
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
PF_OWNER	VARCHAR2(30)	NOT NULL	Owner of the policy function
PACKAGE	VARCHAR2(30)		Name of the package containing the policy function
FUNCTION	VARCHAR2(30)	NOT NULL	Name of the policy function
SEL	VARCHAR2(3)		Indicates whether the policy is applied to queries on the object (YES) or not (NO)
INS	VARCHAR2(3)		Indicates whether the policy is applied to INSERT statements on the object (YES) or not (NO)
UPD	VARCHAR2(3)		Indicates whether the policy is applied to UPDATE statements on the object (YES) or not (NO)
DEL	VARCHAR2(3)		Indicates whether the policy is applied to DELETE statements on the object (YES) or not (NO)
IDX	VARCHAR2(3)		Indicates whether the policy is enforced for index maintenance on the object (YES) or not (NO)
CHK_OPTION	VARCHAR2(3)		Indicates whether the check option is enforced for the policy (YES) or not (NO)
ENABLE	VARCHAR2(3)		Indicates whether the policy is enabled (YES) or disabled (NO)
STATIC_POLICY	VARCHAR2(3)		Indicates whether the policy is static (YES) or not (NO)
POLICY_TYPE	VARCHAR2(24)		Policy type: <ul style="list-style-type: none"> ■ STATIC ■ SHARED_STATIC ■ CONTEXT_SENSITIVE ■ SHARED_CONTEXT_SENSITIVE ■ DYNAMIC
LONG_PREDICATE	VARCHAR2(3)		Indicates whether the policy function can return a maximum of 32 KB of predicate (YES) or not (NO). If NO, the default maximum predicate size is 4000 bytes.

See Also:

- ["DBA_POLICIES"](#) on page 3-100
- ["USER_POLICIES"](#) on page 3-174
- *Oracle Database Concepts* for information on security policies and fine-grained access control
- The DBMS_RLS package in *PL/SQL Packages and Types Reference* for information on administering security policies

ALL_POLICY_CONTEXTS

ALL_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views accessible to the current user.

Related Views

- DBA_POLICY_CONTEXTS describes all driving contexts in the database.
- USER_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
NAMESPACE	VARCHAR2(30)	NOT NULL	Namespace of the driving context
ATTRIBUTE	VARCHAR2(30)	NOT NULL	Attribute of the driving context

See Also:

- ["DBA_POLICY_CONTEXTS"](#) on page 3-100
- ["USER_POLICY_CONTEXTS"](#) on page 3-174

ALL_POLICY_GROUPS

ALL_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views accessible to the current user.

Related Views

- DBA_POLICY_GROUPS describes all policy groups in the database.
- USER_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
POLICY_GROUP	VARCHAR2(30)	NOT NULL	Name of the policy group

See Also:

- ["DBA_POLICY_GROUPS"](#) on page 3-100
- ["USER_POLICY_GROUPS"](#) on page 3-174

ALL_PROCEDURES

ALL_PROCEDURES lists all functions and procedures, along with associated properties. For example, ALL_PROCEDURES indicates whether or not a function is pipelined, parallel enabled or an aggregate function. If a function is pipelined or an aggregate function, the associated implementation type (if any) is also identified.

Related Views

- DBA_PROCEDURES lists all functions and procedures, along with associated properties.
- USER_PROCEDURES lists all functions and procedures, along with associated properties. It does not contain the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the procedure
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object: top-level function, procedure, or package name
PROCEDURE_NAME	VARCHAR2(30)		Name of the procedure
AGGREGATE	VARCHAR2(3)		Indicates whether the procedure is an aggregate function (YES) or not (NO)
PIPELINED	VARCHAR2(3)		Indicates whether the procedure is a pipelined table function (YES) or not (NO)
IMPLTYPEOWNER	VARCHAR2(30)		Name of the owner of the implementation type, if any
IMPLTYPENAME	VARCHAR2(30)		Name of the implementation type, if any
PARALLEL	VARCHAR2(3)		Indicates whether the procedure or function is parallel-enabled (YES) or not (NO)

ALL_PROPAGATION

ALL_PROPAGATION displays information about the Streams propagations that have a source queue accessible to the current user.

Related View

DBA_PROPAGATION displays information about all Streams propagations in the database.

Column	Datatype	NULL	Description
PROPAGATION_NAME	VARCHAR2(30)	NOT NULL	Name of the Streams propagation
SOURCE_QUEUE_OWNER	VARCHAR2(30)		Owner of the source queue of the propagation
SOURCE_QUEUE_NAME	VARCHAR2(30)		Name of the source queue of the propagation
DESTINATION_QUEUE_OWNER	VARCHAR2(30)		Owner of the destination queue of the propagation
DESTINATION_QUEUE_NAME	VARCHAR2(30)		Name of the destination queue of the propagation
DESTINATION_DBLINK	VARCHAR2(128)		Database link to propagate events from the source queue to the destination queue

Column	Datatype	NULL	Description
RULE_SET_OWNER	VARCHAR2(30)		Owner of the propagation positive rule set
RULE_SET_NAME	VARCHAR2(30)		Name of the propagation positive rule set
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the propagation negative rule set
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the propagation negative rule set

See Also: ["DBA_PROPAGATION"](#) on page 3-101

ALL_PUBLISHED_COLUMNS

ALL_PUBLISHED_COLUMNS describes the published source table columns accessible to the current user. This view is intended for use by Change Data Capture subscribers. This view does not display the CHANGE_TABLE_SCHEMA or CHANGE_TABLE_NAME columns.

Related Views

- DBA_PUBLISHED_COLUMNS describes all published source table columns in the database.
- USER_PUBLISHED_COLUMNS describes the published source table columns owned by the current user. This view does not display the CHANGE_TABLE_SCHEMA or CHANGE_TABLE_NAME columns.

Column	Datatype	NULL	Description
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the change set
CHANGE_TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Associated change table owner
CHANGE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Associated change table name
PUB_ID	NUMBER	NOT NULL	Publication identifier
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Table owner in the source database
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name in the source database
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Column datatype
DATA_LENGTH	NUMBER	NOT NULL	Column length (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype; NULL for all other datatypes
DATA_SCALE	NUMBER		Digits to the right of a decimal point in a number
NULLABLE	VARCHAR2(1)		Indicates whether nulls are allowed (Y) or not (N)

See Also:

- ["DBA_PUBLISHED_COLUMNS"](#) on page 3-102
- ["USER_PUBLISHED_COLUMNS"](#) on page 3-175

ALL_QUEUE_TABLES

ALL_QUEUE_TABLES describes the queues in the queue tables accessible to the current user.

Related Views

- `DBA_QUEUE_TABLES` describes the queues in all queue tables in the database.
- `USER_QUEUE_TABLES` describes the queues in the queue tables created in the current user's schema. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Schema of the queue table
<code>QUEUE_TABLE</code>	<code>VARCHAR2(30)</code>		Name of the queue table
<code>TYPE</code>	<code>VARCHAR2(7)</code>		Type of user data: <ul style="list-style-type: none"> ■ <code>RAW</code> - Raw type ■ <code>OBJECT</code> - User-defined object type ■ <code>VARIANT</code> - Variant type (internal use only)
<code>OBJECT_TYPE</code>	<code>VARCHAR2(61)</code>		Object type of the payload when <code>TYPE</code> is <code>OBJECT</code>
<code>SORT_ORDER</code>	<code>VARCHAR2(22)</code>		User-specified sort order
<code>RECIPIENTS</code>	<code>VARCHAR2(8)</code>		<code>SINGLE</code> or <code>MULTIPLE</code> recipients
<code>MESSAGE_GROUPING</code>	<code>VARCHAR2(13)</code>		<code>NONE</code> or <code>TRANSACTIONAL</code>
<code>COMPATIBLE</code>	<code>VARCHAR2(5)</code>		Lowest release level which the queue table is compatible with (for example, 8.0.3)
<code>PRIMARY_INSTANCE</code>	<code>NUMBER</code>		Indicates the instance number of the instance which is the primary owner of the queue table. A value of 0 indicates that there is no primary owner.
<code>SECONDARY_INSTANCE</code>	<code>NUMBER</code>		Indicates the instance number of the instance which is the secondary owner of the queue table. This instance becomes the owner of the queue table if the primary owner is not alive. A value of 0 indicates that there is no secondary owner.
<code>OWNER_INSTANCE</code>	<code>NUMBER</code>		Instance number of the instance which currently owns the queue table
<code>USER_COMMENT</code>	<code>VARCHAR2(50)</code>		Comment supplied by the user
<code>SECURE</code>	<code>VARCHAR2(3)</code>		Indicates whether the queue table is secure (<code>YES</code>) or not (<code>NO</code>)

See Also:

- ["DBA_QUEUE_TABLES"](#) on page 3-103
- ["USER_QUEUE_TABLES"](#) on page 3-175
- *Oracle Streams Advanced Queuing User's Guide and Reference* for more information about these views and Advanced Queuing

ALL_QUEUES

`ALL_QUEUES` describes all queues on which the current user has enqueue or dequeue privileges. If the user has any Advanced Queuing system privileges, like `MANAGE ANY QUEUE`, `ENQUEUE ANY QUEUE` or `DEQUEUE ANY QUEUE`, then this view describes all queues in the database.

Related Views

- `DBA_QUEUES` describes all queues in the database.
- `USER_QUEUES` describes the operational characteristics of every queue owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the table the queue data resides in
QID	NUMBER	NOT NULL	Object number of the queue
QUEUE_TYPE	VARCHAR2(15)		Type of the queue
MAX_RETRIES	NUMBER		Maximum number of retries allowed when dequeuing from the queue
RETRY_DELAY	NUMBER		Time interval between retries
ENQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for enqueue
DEQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for dequeue
RETENTION	VARCHAR2(40)		Time interval processed messages retained in the queue
USER_COMMENT	VARCHAR2(50)		User specified comment

See Also: *Oracle Streams Advanced Queuing User's Guide and Reference* for more information about these views and Advanced Queuing

ALL_REFRESH

ALL_REFRESH describes all the refresh groups accessible to the current user.

Related Views

- DBA_REFRESH describes all refresh groups in the database.
- USER_REFRESH describes all refresh groups owned by the current user.

Column	Datatype	NULL	Description
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		(Y N) If Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y N) If Y then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		(Y N) Y means the job is broken and will never be run
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

ALL_REFRESH_CHILDREN

ALL_REFRESH_CHILDREN lists all the objects in refresh groups that are accessible to the current user.

Related Views

- DBA_REFRESH_CHILDREN describes the objects in all refresh groups in the database.
- USER_REFRESH_CHILDREN describes the objects in all refresh groups owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		(Y N) If Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y N) If Y then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		(Y N) Y means the job is broken and will never be run
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

ALL_REFRESH_DEPENDENCIES

ALL_REFRESH_DEPENDENCIES lists the names of the dependent detail or container tables of all the materialized views in the current schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name, unique within this schema
PARENT_OBJECT_TYPE	CHAR(17)		MATERIALIZED VIEW
OLDEST_REFRESH_SCN	NUMBER		The minimum SCN of any summary or materialized view that has TABLE_NAME as a detail table
OLDEST_REFRESH_DATE	DATE		SYSDATE when last refreshed

ALL_REFS

ALL_REFS describes the REF columns and REF attributes in object type columns accessible to the current user.

Related Views

- DBA_REFS describes all REF columns and REF attributes in the database.
- USER_REFS describes the REF columns and REF attributes in object type columns owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the REF column or attribute. If it is not a top-level attribute, the value of COLUMN_NAME should be a path name starting with the column name
WITH_ROWID	VARCHAR2(3)		Whether the REF value is stored with ROWID (YES NO)?
IS_SCOPED	VARCHAR2(3)		Whether the REF column is scoped (YES NO)?
SCOPE_TABLE_OWNER	VARCHAR2(30)		Name of the owner of the scope table, if it exists and is accessible by the user
SCOPE_TABLE_NAME	VARCHAR2(30)		Name of the scope table, if it exists and is accessible by the user
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED

ALL_REGISTERED_MVIEWS

ALL_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) accessible to the current user.

Related Views

- DBA_REGISTERED_MVIEWS describes all registered materialized views in the database.
- USER_REGISTERED_MVIEWS describes all registered materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MVIEW_SITE	VARCHAR2(128)	NOT NULL	Global name of the materialized view site
CAN_USE_LOG	VARCHAR2(3)		YES if the materialized view can use a materialized view log, NO if the materialized view is too complex to use a log.
UPDATABLE	VARCHAR2(3)		(YES/NO) Indicates whether the materialized view is updatable. If set to NO, then the materialized view is read only.
REFRESH_METHOD	VARCHAR2(11)		Whether the materialized view uses primary key, rowids, or object identifiers for fast refresh
MVIEW_ID	NUMBER(38)		Identifier for the materialized view used by the masters for fast refresh

Column	Datatype	NULL	Description
VERSION	VARCHAR2(26)		Oracle version of the materialized view Note: Oracle Database materialized views show ORACLE 8 MATERIALIZED VIEW.
QUERY_TXT	LONG		Query that defines the materialized view

See Also:

- ["DBA_REGISTERED_MVIEWS"](#) on page 3-107
- ["USER_REGISTERED_MVIEWS"](#) on page 3-176

ALL_REGISTRY_BANNERS

ALL_REGISTRY_BANNERS displays the valid components loaded into the database.

Column	Datatype	NULL	Description
BANNER	VARCHAR2(80)		Component display banner

ALL_REWRITE_EQUIVALENCES

ALL_REWRITE_EQUIVALENCES describes the rewrite equivalences accessible to the current user.

Related Views

- DBA_REWRITE_EQUIVALENCES describes all rewrite equivalences in the database.
- USER_REWRITE_EQUIVALENCES describes the rewrite equivalences owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the rewrite equivalence
NAME	VARCHAR2(30)	NOT NULL	Name of the rewrite equivalence
SOURCE_STMT	CLOB		Source statement of the rewrite equivalence
DESTINATION_STMT	CLOB		Destination of the rewrite equivalence
REWRITE_MODE	VARCHAR2(10)		Rewrite mode of the rewrite equivalence: <ul style="list-style-type: none"> ▪ DISABLED ▪ TEXT_MATCH ▪ GENERAL ▪ RECURSIVE

See Also:

- ["DBA_REWRITE_EQUIVALENCES"](#) on page 3-110
- ["USER_REWRITE_EQUIVALENCES"](#) on page 3-176

ALL_RULE_SET_RULES

ALL_RULE_SET_RULES describes the rules in the rule sets accessible to the current user.

Related Views

- [DBA_RULE_SET_RULES](#) describes the rules in all rule sets in the database.
- [USER_RULE_SET_RULES](#) describes the rules in the rule sets owned by the current user. This view does not display the `RULE_SET_OWNER` column.

Column	Datatype	NULL	Description
<code>RULE_SET_OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the rule set
<code>RULE_SET_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the rule set
<code>RULE_OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the rule
<code>RULE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the rule
<code>RULE_SET_RULE_ENABLED</code>	<code>VARCHAR2(8)</code>		Indicates whether the rule is enabled in the rule set (ENABLED) or not (DISABLED)
<code>RULE_SET_RULE_EVAL_CTX_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the evaluation context specified when the rule was added to the rule set, if any
<code>RULE_SET_RULE_EVAL_CTX_NAME</code>	<code>VARCHAR2(30)</code>		Name of the evaluation context specified when the rule was added to the rule set, if any
<code>RULE_SET_RULE_COMMENT</code>	<code>VARCHAR2(4000)</code>		Comment specified when the rule was added to the rule set, if any

See Also:

- ["DBA_RULE_SET_RULES"](#) on page 3-115
- ["USER_RULE_SET_RULES"](#) on page 3-177

ALL_RULE_SETS

`ALL_RULE_SETS` describes the rule sets accessible to the current user.

Related Views

- [DBA_RULE_SETS](#) describes all rule sets in the database.
- [USER_RULE_SETS](#) describes the rule sets owned by the current user. This view does not display the `RULE_SET_OWNER` column.

Column	Datatype	NULL	Description
<code>RULE_SET_OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the rule set
<code>RULE_SET_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the rule set
<code>RULE_SET_EVAL_CONTEXT_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the evaluation context associated with the rule set, if any
<code>RULE_SET_EVAL_CONTEXT_NAME</code>	<code>VARCHAR2(30)</code>		Name of the evaluation context associated with the rule set, if any
<code>RULE_SET_COMMENT</code>	<code>VARCHAR2(4000)</code>		Comment specified with the rule set, if any

See Also:

- ["DBA_RULE_SETS"](#) on page 3-115
- ["USER_RULE_SETS"](#) on page 3-177

ALL_RULES

`ALL_RULES` describes the rules accessible to the current user.

Related Views

- `DBA_RULES` describes all rules in the database.
- `USER_RULES` describes the rules owned by the current user. This view does not display the `RULE_OWNER` column.

Column	Datatype	NULL	Description
<code>RULE_OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the rule
<code>RULE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the rule
<code>RULE_CONDITION</code>	<code>CLOB</code>		Expressions and operators that constitute the rule condition
<code>RULE_EVALUATION_CONTEXT_OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the evaluation context associated with the rule, if any
<code>RULE_EVALUATION_CONTEXT_NAME</code>	<code>VARCHAR2(30)</code>		Name of the evaluation context associated with the rule, if any
<code>RULE_ACTION_CONTEXT</code>	<code>RE\$NV_LIST</code>		Action context associated with the rule, if any
<code>RULE_COMMENT</code>	<code>VARCHAR2(4000)</code>		Comment specified with the rule, if any

See Also:

- ["DBA_RULES"](#) on page 3-115
- ["USER_RULES"](#) on page 3-177

ALL_SCHEDULER_JOB_ARGS

`ALL_SCHEDULER_JOB_ARGS` displays information about the arguments of the Scheduler jobs accessible to the current user.

Related Views

- `DBA_SCHEDULER_JOB_ARGS` displays information about the arguments of all Scheduler jobs in the database.
- `USER_SCHEDULER_JOB_ARGS` displays information about the arguments of the Scheduler jobs owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the job to which the argument belongs
<code>JOB_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the job to which the argument belongs
<code>ARGUMENT_NAME</code>	<code>VARCHAR2(30)</code>		Optional name of the argument
<code>ARGUMENT_POSITION</code>	<code>NUMBER</code>		Position of the argument in the argument list
<code>ARGUMENT_TYPE</code>	<code>VARCHAR2(61)</code>		Datatype of the argument
<code>VALUE</code>	<code>VARCHAR2(4000)</code>		Value of the argument (in string format) if the argument is a string
<code>ANYDATA_VALUE</code>	<code>ANYDATA</code>		Value of the argument (in AnyData format)
<code>OUT_ARGUMENT</code>	<code>VARCHAR2(5)</code>		Reserved for future use

See Also:

- ["DBA_SCHEDULER_JOB_ARGS"](#) on page 3-115
- ["USER_SCHEDULER_JOB_ARGS"](#) on page 3-177

ALL_SCHEDULER_JOB_CLASSES

ALL_SCHEDULER_JOB_CLASSES displays information about the Scheduler job classes accessible to the current user.

Related View

DBA_SCHEDULER_JOB_CLASSES displays information about all Scheduler job classes in the database.

Column	Datatype	NULL	Description
JOB_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler job class
RESOURCE_CONSUMER_GROUP	VARCHAR2(30)		Resource consumer group associated with the class
SERVICE	VARCHAR2(64)		Name of the service the class is affined with
LOGGING_LEVEL	VARCHAR2(4)		Amount of logging that will be done pertaining to the class: <ul style="list-style-type: none"> ▪ OFF ▪ RUNS ▪ FULL
LOG_HISTORY	NUMBER		History to maintain in the job log (in days) for the class
COMMENTS	VARCHAR2(240)		Comments on the class

See Also: ["DBA_SCHEDULER_JOB_CLASSES"](#) on page 3-115

ALL_SCHEDULER_JOB_LOG

ALL_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOB_LOG displays log information for all Scheduler jobs in the database.
- USER_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs owned by the current user.

Column	Datatype	NULL	Description
LOG_ID	NUMBER	NOT NULL	Unique identifier that identifies a row
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
OWNER	VARCHAR2(30)		Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)		Name of the Scheduler job
JOB_CLASS	VARCHAR2(30)		Class that the job belonged to at the time of entry
OPERATION	VARCHAR2(30)		Operation corresponding to the log entry
STATUS	VARCHAR2(30)		Status of the operation, if applicable
USER_NAME	VARCHAR2(30)		Name of the user who performed the operation, if applicable
CLIENT_ID	VARCHAR2(64)		Client identifier of the user who performed the operation, if applicable
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user who performed the operation, if applicable

Column	Datatype	NULL	Description
ADDITIONAL_INFO	CLOB		Additional information on the entry, if applicable

See Also:

- ["DBA_SCHEDULER_JOB_LOG"](#) on page 3-115
- ["USER_SCHEDULER_JOB_LOG"](#) on page 3-177

ALL_SCHEDULER_JOB_RUN_DETAILS

ALL_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOB_RUN_DETAILS displays log run details for all Scheduler jobs in the database.
- USER_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs owned by the current user.

Column	Datatype	NULL	Description
LOG_ID	NUMBER		Unique identifier of the log entry
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
OWNER	VARCHAR2(30)		Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)		Name of the Scheduler job
STATUS	VARCHAR2(30)		Status of the job run
ERROR#	NUMBER		Error number in the case of an error
REQ_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Requested start date of the job run
ACTUAL_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Actual date on which the job was run
RUN_DURATION	INTERVAL DAY(3) TO SECOND(0)		Duration of the job run
INSTANCE_ID	NUMBER		Identifier of the instance on which the job was run
SESSION_ID	VARCHAR2(30)		Session identifier of the job run
SLAVE_PID	VARCHAR2(30)		Process identifier of the slave on which the job was run
CPU_USED	NUMBER		Amount of CPU used for the job run
ADDITIONAL_INFO	VARCHAR2(4000)		Additional information on the job run, if applicable

See Also:

- ["DBA_SCHEDULER_JOB_RUN_DETAILS"](#) on page 3-115
- ["USER_SCHEDULER_JOB_RUN_DETAILS"](#) on page 3-178

ALL_SCHEDULER_JOBS

ALL_SCHEDULER_JOBS displays information about the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOBS displays information about all Scheduler jobs in the database.
- USER_SCHEDULER_JOBS displays information about the Scheduler jobs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler job
JOB_CREATOR	VARCHAR2(30)		Original creator of the job
CLIENT_ID	VARCHAR2(64)		Client identifier of the user creating the job
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user creating the job
PROGRAM_OWNER	VARCHAR2(4000)		Owner of the program associated with the job
PROGRAM_NAME	VARCHAR2(4000)		Name of the program associated with the job
JOB_TYPE	VARCHAR2(16)		Inline job action type: <ul style="list-style-type: none"> ■ PLSQL_BLOCK ■ STORED_PROCEDURE ■ EXECUTABLE
JOB_ACTION	VARCHAR2(4000)		Inline job action
NUMBER_OF_ARGUMENTS	NUMBER		Inline number of job arguments
SCHEDULE_OWNER	VARCHAR2(4000)		Owner of the job's schedule (can be a window or a window group)
SCHEDULE_NAME	VARCHAR2(4000)		Name of the job's schedule (can be a window or a window group)
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Original scheduled start date of the job (for an inline schedule)
REPEAT_INTERVAL	VARCHAR2(4000)		Inline schedule PL/SQL expression or calendar string
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Date after which the job will no longer run (for an inline schedule)
JOB_CLASS	VARCHAR2(30)		Name of the job class associated with the job
ENABLED	VARCHAR2(5)		Indicates whether the job is enabled (TRUE) or not (FALSE)
AUTO_DROP	VARCHAR2(5)		Indicates whether the job will be dropped when it has completed (TRUE) or not (FALSE)
RESTARTABLE	VARCHAR2(5)		Indicates whether the job can be restarted (TRUE) or not (FALSE)
STATE	VARCHAR2(15)		Current state of the job: <ul style="list-style-type: none"> ■ DISABLED ■ RETRY SCHEDULED ■ SCHEDULED ■ RUNNING ■ COMPLETED ■ BROKEN ■ FAILED ■ REMOTE ■ SUCCEEDED
JOB_PRIORITY	NUMBER		Priority of the job relative to other jobs in the same class
RUN_COUNT	NUMBER		Number of times the job has run
MAX_RUNS	NUMBER		Maximum number of times the job is scheduled to run

Column	Datatype	NULL	Description
FAILURE_COUNT	NUMBER		Number of times the job has failed to run
MAX_FAILURES	NUMBER		Number of times the job will be allowed to fail before being marked broken
RETRY_COUNT	NUMBER		Number of times the job has retried, if it is retrying
LAST_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Last date on which the job started running
LAST_RUN_DURATION	INTERVAL DAY(9) TO SECOND(6)		Amount of time the job took to complete during the last run
NEXT_RUN_DATE	TIMESTAMP(6) WITH TIME ZONE		Next date on which the job is scheduled to run
SCHEDULE_LIMIT	INTERVAL DAY(3) TO SECOND(0)		Time after which a job which has not run yet will be rescheduled
MAX_RUN_DURATION	INTERVAL DAY(3) TO SECOND(0)		Maximum amount of time for which the job will be allowed to run
LOGGING_LEVEL	VARCHAR2(4)		Amount of logging that will be done pertaining to the job: <ul style="list-style-type: none"> ▪ OFF ▪ RUNS ▪ FULL
STOP_ON_WINDOW_CLOSE	VARCHAR2(5)		Indicates whether the job will stop if a window associated with the job closes (TRUE) or not (FALSE)
INSTANCE_STICKINESS	VARCHAR2(5)		Indicates whether the job is sticky (TRUE) or not (FALSE)
SYSTEM	VARCHAR2(5)		Indicates whether the job is a system job (TRUE) or not (FALSE)
JOB_WEIGHT	NUMBER		Weight of the job
NLS_ENV	VARCHAR2(4000)		NLS environment of the job
SOURCE	VARCHAR2(128)		Source global database identifier
DESTINATION	VARCHAR2(128)		Destination global database identifier
COMMENTS	VARCHAR2(240)		Comments on the job
FLAGS	NUMBER		This column is for internal use.

See Also:

- ["DBA_SCHEDULER_JOBS"](#) on page 3-115
- ["USER_SCHEDULER_JOBS"](#) on page 3-178

ALL_SCHEDULER_PROGRAM_ARGS

ALL_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs accessible to the current user.

Related Views

- DBA_SCHEDULER_PROGRAM_ARGS displays information about the arguments of all Scheduler programs in the database.
- USER_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the program to which the argument belongs
PROGRAM_NAME	VARCHAR2(30)	NOT NULL	Name of the program to which the argument belongs
ARGUMENT_NAME	VARCHAR2(30)		Optional name of the argument
ARGUMENT_POSITION	NUMBER	NOT NULL	Position of the argument in the argument list
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the argument
METADATA_ATTRIBUTE	VARCHAR2(12)		Metadata attribute: <ul style="list-style-type: none"> ▪ JOB_NAME ▪ JOB_OWNER ▪ JOB_START ▪ WINDOW_START ▪ WINDOW_END
DEFAULT_VALUE	VARCHAR2(4000)		Default value taken by the argument (in string format) if the argument is a string
DEFAULT_ANYDATA_VALUE	ANYDATA		Default value taken by the argument (in AnyData format)
OUT_ARGUMENT	VARCHAR2(5)		Reserved for future use

See Also:

- ["DBA_SCHEDULER_PROGRAM_ARGS"](#) on page 3-116
- ["USER_SCHEDULER_PROGRAM_ARGS"](#) on page 3-178

ALL_SCHEDULER_PROGRAMS

ALL_SCHEDULER_PROGRAMS displays information about the Scheduler programs accessible to the current user.

Related Views

- DBA_SCHEDULER_PROGRAMS displays information about all Scheduler programs in the database.
- USER_SCHEDULER_PROGRAMS displays information about the Scheduler programs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler program
PROGRAM_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler program
PROGRAM_TYPE	VARCHAR2(16)		Program action type: <ul style="list-style-type: none"> ▪ PLSQL_BLOCK ▪ STORED_PROCEDURE ▪ EXECUTABLE
PROGRAM_ACTION	VARCHAR2(4000)		Program action
NUMBER_OF_ARGUMENTS	NUMBER		Number of arguments accepted by the program
ENABLED	VARCHAR2(5)		Indicates whether the program is enabled (TRUE) or not (FALSE)
COMMENTS	VARCHAR2(240)		Comments on the program

See Also:

- ["DBA_SCHEDULER_PROGRAMS"](#) on page 3-116
- ["USER_SCHEDULER_PROGRAMS"](#) on page 3-178

ALL_SCHEDULER_RUNNING_JOBS

ALL_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_RUNNING_JOBS displays information about all running Scheduler jobs in the database.
- USER_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the running Scheduler job
JOB_NAME	VARCHAR2(30)	NOT NULL	Name of the running Scheduler job
SESSION_ID	NUMBER		Identifier of the session running the Scheduler job
SLAVE_PROCESS_ID	NUMBER		Process number of the slave process running the Scheduler job
RUNNING_INSTANCE	NUMBER		Database instance number of the slave process running the Scheduler job
RESOURCE_CONSUMER_GROUP	VARCHAR2(32)		Resource consumer group of the session in which the Scheduler job is running
ELAPSED_TIME	INTERVAL DAY(3) TO SECOND(2)		Elapsed time since the Scheduler job was started
CPU_USED	NUMBER		Scheduler job CPU time consumed, if available

See Also:

- ["DBA_SCHEDULER_RUNNING_JOBS"](#) on page 3-116
- ["USER_SCHEDULER_RUNNING_JOBS"](#) on page 3-178

ALL_SCHEDULER_SCHEDULES

ALL_SCHEDULER_SCHEDULES displays information about the Scheduler schedules accessible to the current user.

Related Views

- DBA_SCHEDULER_SCHEDULES displays information about all Scheduler schedules in the database.
- USER_SCHEDULER_SCHEDULES displays information about the Scheduler schedules owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the schedule

Column	Datatype	NULL	Description
SCHEDULE_NAME	VARCHAR2(30)	NOT NULL	Name of the schedule
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Start date for the repeat interval
REPEAT_INTERVAL	VARCHAR2(4000)		Calendar syntax expression for the schedule
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Cutoff date after which the schedule will not specify any dates
COMMENTS	VARCHAR2(240)		Comments on the schedule

See Also:

- ["DBA_SCHEDULER_SCHEDULES"](#) on page 3-116
- ["USER_SCHEDULER_SCHEDULES"](#) on page 3-178

ALL_SCHEDULER_WINDOW_DETAILS

ALL_SCHEDULER_WINDOW_DETAILS displays log details for the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_DETAILS displays log details for all Scheduler windows in the database.

Column	Datatype	NULL	Description
LOG_ID	NUMBER		Unique identifier of the log entry
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
WINDOW_NAME	VARCHAR2(30)		Name of the Scheduler window
REQ_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Requested start date for the Scheduler window
ACTUAL_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Actual start date of the Scheduler window
WINDOW_DURATION	INTERVAL DAY(3) TO SECOND(0)		Requested duration of the Scheduler window
ACTUAL_DURATION	INTERVAL DAY(3) TO SECOND(0)		Actual duration for which the Scheduler window lasted
INSTANCE_ID	NUMBER		Identifier of the instance on which the window was run
ADDITIONAL_INFO	VARCHAR2(4000)		Additional information on the entry, if applicable

See Also: ["DBA_SCHEDULER_WINDOW_DETAILS"](#) on page 3-116

ALL_SCHEDULER_WINDOW_GROUPS

ALL_SCHEDULER_WINDOW_GROUPS displays information about the Scheduler window groups accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_GROUPS displays information about all Scheduler window groups in the database.

Column	Datatype	NULL	Description
WINDOW_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the window group
ENABLED	VARCHAR2(5)		Indicates whether the window group is enabled (TRUE) or not (FALSE)
NUMBER_OF_WINDOWS	NUMBER		Number of members in the window group
COMMENTS	VARCHAR2(240)		Optional comment about the window group

See Also: ["DBA_SCHEDULER_WINDOW_GROUPS"](#) on page 3-116

ALL_SCHEDULER_WINDOW_LOG

ALL_SCHEDULER_WINDOW_LOG displays log information for the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_LOG displays log information for all Scheduler windows in the database.

Column	Datatype	NULL	Description
LOG_ID	NUMBER	NOT NULL	Unique identifier of the log entry
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
WINDOW_NAME	VARCHAR2(30)		Name of the Scheduler window
OPERATION	VARCHAR2(30)		Operation corresponding to the log entry
STATUS	VARCHAR2(30)		Status of the operation, if applicable
USER_NAME	VARCHAR2(30)		Name of the user who performed the operation, if applicable
CLIENT_ID	VARCHAR2(64)		Client identifier of the user who performed the operation, if applicable
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user who performed the operation, if applicable
ADDITIONAL_INFO	CLOB		Additional information on the entry, if applicable

See Also: ["DBA_SCHEDULER_WINDOW_LOG"](#) on page 3-116

ALL_SCHEDULER_WINDOWS

ALL_SCHEDULER_WINDOWS displays information about the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOWS displays information about all Scheduler windows in the database.

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler window
RESOURCE_PLAN	VARCHAR2(30)		Resource plan associated with the window
SCHEDULE_OWNER	VARCHAR2(4000)		Owner of the window's schedule
SCHEDULE_NAME	VARCHAR2(4000)		Name of the window's schedule

Column	Datatype	NULL	Description
START_DATE	TIMESTAMP (6) WITH TIME ZONE		Start date of the window (for an inline schedule)
REPEAT_INTERVAL	VARCHAR2 (4000)		Calendar string for the window (for an inline schedule)
END_DATE	TIMESTAMP (6) WITH TIME ZONE		Date after which the window will no longer open (for an inline schedule)
DURATION	INTERVAL DAY (3) TO SECOND (0)		Duration of the window
WINDOW_PRIORITY	VARCHAR2 (4)		Priority of the job relative to other windows: <ul style="list-style-type: none"> ■ HIGH ■ LOW
NEXT_START_DATE	TIMESTAMP (6) WITH TIME ZONE		Next date on which the window is scheduled to start
LAST_START_DATE	TIMESTAMP (6) WITH TIME ZONE		Last date on which the window opened
ENABLED	VARCHAR2 (5)		Indicates whether the window is enabled (TRUE) or disabled (FALSE)
ACTIVE	VARCHAR2 (5)		Indicates whether the window is open (TRUE) or not (FALSE)
COMMENTS	VARCHAR2 (240)		Comments on the window

See Also: ["DBA_SCHEDULER_WINDOWS"](#) on page 3-117

ALL_SCHEDULER_WINGROUP_MEMBERS

ALL_SCHEDULER_WINGROUP_MEMBERS displays the members of the Scheduler window groups accessible to the current user.

Related View

DBA_SCHEDULER_WINGROUP_MEMBERS displays the members of all Scheduler window groups in the database.

Column	Datatype	NULL	Description
WINDOW_GROUP_NAME	VARCHAR2 (30)	NOT NULL	Name of the window group
WINDOW_NAME	VARCHAR2 (30)	NOT NULL	Name of the window member of the window group

See Also: ["DBA_SCHEDULER_WINGROUP_MEMBERS"](#) on page 3-117

ALL_SEC_RELEVANT_COLS

ALL_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views accessible to the current user.

Related Views

- DBA_SEC_RELEVANT_COLS describes the security relevant columns of all security policies in the database.
- USER_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or view
OBJECT_NAME	VARCHAR2(30)		Name of the table or view
POLICY_GROUP	VARCHAR2(30)		Name of the policy group
POLICY_NAME	VARCHAR2(30)		Name of the policy
SEC_REL_COLUMN	VARCHAR2(30)		Name of the security relevant column
COLUMN_OPTION	VARCHAR2(8)		Option of the security relevant column: <ul style="list-style-type: none"> ■ NONE ■ ALL_ROWS

See Also:

- ["DBA_SEC_RELEVANT_COLS"](#) on page 3-117
- ["USER_SEC_RELEVANT_COLS"](#) on page 3-178

ALL_SEQUENCES

ALL_SEQUENCES describes all sequences accessible to the current user.

Related Views

- DBA_SEQUENCES describes all sequences in the database.
- USER_SEQUENCES describes all sequences owned by the current user. This view does not display the SEQUENCE_OWNER column.

Column	Datatype	NULL	Description
SEQUENCE_OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the sequence
SEQUENCE_NAME	VARCHAR2(30)	NOT NULL	Sequence name
MIN_VALUE	NUMBER		Minimum value of the sequence
MAX_VALUE	NUMBER		Maximum value of the sequence
INCREMENT_BY	NUMBER	NOT NULL	Value by which sequence is incremented
CYCLE_FLAG	VARCHAR2(1)		Does sequence wrap around on reaching limit
ORDER_FLAG	VARCHAR2(1)		Are sequence numbers generated in order
CACHE_SIZE	NUMBER	NOT NULL	Number of sequence numbers to cache
LAST_NUMBER	NUMBER	NOT NULL	Last sequence number written to disk. If a sequence uses caching, the number written to disk is the last number placed in the sequence cache. This number is likely to be greater than the last sequence number that was used.

ALL_SERVICES

ALL_SERVICES displays all services in the database. The view excludes rows marked for deletion.

Related View

DBA_SERVICES displays all services in the database. The view excludes rows marked for deletion.

ALL_SOURCE

Column	Datatype	NULL	Description
SERVICE_ID	NUMBER		Unique ID for the service
NAME	VARCHAR2(64)		Short name for the service
NAME_HASH	NUMBER		Hash of the short name for the service
NETWORK_NAME	VARCHAR2(512)		Network name used to connect to the service
CREATION_DATE	DATE		Date the service was created
CREATION_DATE_HASH	NUMBER		Hash of the creation date
FAILOVER_METHOD	VARCHAR2(64)		Failover method (BASIC or NONE) for the service
FAILOVER_TYPE	VARCHAR2(64)		Failover type (SESSION or SELECT) for the service
FAILOVER_RETRIES	NUMBER(10)		Number of retries when failing over the service
FAILOVER_DELAY	NUMBER(10)		Delay between retries when failing over the service

ALL_SOURCE

ALL_SOURCE describes the text source of the stored objects accessible to the current user.

Related Views

- DBA_SOURCE describes the text source of all stored objects in the database.
- USER_SOURCE describes the text source of the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: FUNCTION, JAVA_SOURCE, PACKAGE, PACKAGE_BODY, PROCEDURE, TRIGGER, TYPE, TYPE_BODY
LINE	NUMBER	NOT NULL	Line number of this line of source
TEXT	VARCHAR2(4000)		Text source of the stored object

ALL_SOURCE_TABLES

ALL_SOURCE_TABLES describes the existing source tables accessible to the current user. This view is intended for use by Change Data Capture subscribers.

Related Views

- DBA_SOURCE_TABLES describes all existing source tables in the database.
- USER_SOURCE_TABLES describes the existing source tables owned by the current user.

Column	Datatype	NULL	Description
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Table owner in the source database
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name in the source database

See Also:

- ["DBA_SOURCE_TABLES"](#) on page 3-119
- ["USER_SOURCE_TABLES"](#) on page 3-179

ALL_SQLJ_TYPE_ATTRS

ALL_SQLJ_TYPE_ATTRS describes the attributes of the SQLJ object types accessible to the current user.

Related Views

- DBA_SQLJ_TYPE_ATTRS describes the attributes of all SQLJ object types in the database.
- USER_SQLJ_TYPE_ATTRS describes the attributes of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
ATTR_NAME	VARCHAR2(30)	NOT NULL	Name of the attribute
EXTERNAL_ATTR_NAME	VARCHAR2(4000)		External name of the attribute
ATTR_TYPE_MOD	VARCHAR2(7)		Type modifier of the attribute
ATTR_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the attribute
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute, or maximum length of the VARCHAR or VARCHAR2 attribute.
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute, or binary precision of the FLOAT attribute.
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET_NAME	VARCHAR2(44)		Character set name of the attribute (CHAR_CS or NCHAR_CS)
ATTR_NO	NUMBER	NOT NULL	Syntactical order number or position of the attribute as specified in the type specification or CREATE TYPE statement (not to be used as an ID number)
INHERITED	VARCHAR2(3)		Indicates whether the attribute is inherited from a supertype (YES) or not (NO)

See Also:

- ["DBA_SQLJ_TYPE_ATTRS"](#) on page 3-119
- ["USER_SQLJ_TYPE_ATTRS"](#) on page 3-179

ALL_SQLJ_TYPE_METHODS

ALL_SQLJ_TYPE_METHODS describes the methods of the SQLJ object types accessible to the current user.

Related Views

- DBA_SQLJ_TYPE_METHODS describes the methods of all SQLJ object types in the database.

- `USER_SQLJ_TYPE_METHODS` describes the methods of the SQLJ object types owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the type
<code>TYPE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the type
<code>METHOD_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the method
<code>EXTERNAL_VAR_NAME</code>	<code>VARCHAR2(4000)</code>		Name of the external variable
<code>METHOD_NO</code>	<code>NUMBER</code>	NOT NULL	Method number that distinguishes overloaded method (not to be used as an ID number)
<code>METHOD_TYPE</code>	<code>VARCHAR2(6)</code>		Type of the method
<code>PARAMETERS</code>	<code>NUMBER</code>	NOT NULL	Number of parameters to the method
<code>RESULTS</code>	<code>NUMBER</code>	NOT NULL	Number of results returned by the method
<code>FINAL</code>	<code>VARCHAR2(3)</code>		Indicates whether the method is final (YES) or not (NO)
<code>INSTANTIABLE</code>	<code>VARCHAR2(3)</code>		Indicates whether the method is instantiable (YES) or not (NO)
<code>OVERRIDING</code>	<code>VARCHAR2(3)</code>		Indicates whether the method is overriding a supertype method (YES) or not (NO)
<code>INHERITED</code>	<code>VARCHAR2(3)</code>		Indicates whether the method is inherited from a supertype (YES) or not (NO)

See Also:

- ["DBA_SQLJ_TYPE_METHODS"](#) on page 3-119
- ["USER_SQLJ_TYPE_METHODS"](#) on page 3-179

ALL_SQLJ_TYPES

`ALL_SQLJ_TYPES` describes the SQLJ object types accessible to the current user.

Related Views

- `DBA_SQLJ_TYPES` describes all SQLJ object types in the database.
- `USER_SQLJ_TYPES` describes the SQLJ object types owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Owner of the type
<code>TYPE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the type
<code>TYPE_OID</code>	<code>RAW(16)</code>	NOT NULL	Object identifier (OID) of the type
<code>EXTERNAL_NAME</code>	<code>VARCHAR2(4000)</code>		External class name of the type
<code>USING</code>	<code>VARCHAR2(21)</code>		Representation of the type
<code>TYPECODE</code>	<code>VARCHAR2(30)</code>		Typecode of the type
<code>ATTRIBUTES</code>	<code>NUMBER</code>		Number of attributes (if any) in the type
<code>METHODS</code>	<code>NUMBER</code>		Number of methods (if any) in the type
<code>PREDEFINED</code>	<code>VARCHAR2(3)</code>		Indicates whether the type is a predefined type (YES) or not (NO)
<code>INCOMPLETE</code>	<code>VARCHAR2(3)</code>		Indicates whether the type is an incomplete type (YES) or not (NO)

Column	Datatype	NULL	Description
FINAL	VARCHAR2 (3)		Indicates whether the type is a final type (YES) or not (NO)
INSTANTIABLE	VARCHAR2 (3)		Indicates whether the type is an instantiable type (YES) or not (NO)
SUPERTYPE_OWNER	VARCHAR2 (30)		Owner of the supertype (NULL if type is not a subtype)
SUPERTYPE_NAME	VARCHAR2 (30)		Name of the supertype (NULL if type is not a subtype)
LOCAL_ATTRIBUTES	NUMBER		Number of local (not inherited) attributes (if any) in the subtype
LOCAL_METHODS	NUMBER		Number of local (not inherited) methods (if any) in the subtype

See Also:

- ["DBA_SQLJ_TYPES"](#) on page 3-120
- ["USER_SQLJ_TYPES"](#) on page 3-179

ALL_STORED_SETTINGS

ALL_STORED_SETTINGS provides information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges.

Related Views

- DBA_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges. It also returns parameter information for all objects in the database and is accessible only to users with the SELECT_CATALOG_ROLE privilege.
- USER_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units, but only shows information about PL/SQL units owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)	NOT NULL	Name of the database user owning the stored PL/SQL unit
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	Name of the PL/SQL unit
OBJECT_ID	NUMBER	NOT NULL	Object number of the PL/SQL unit
OBJECT_TYPE	VARCHAR2 (12)		The type of PL/SQL unit: PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, TYPE, or TYPE BODY
PARAM_NAME	VARCHAR2 (30)		The name of the parameter stored persistently with the PL/SQL unit
PARAM_VALUE	VARCHAR2 (4000)		The TO_CHAR () representation of the value of the persistently stored parameter. The width of this column is operating system dependent; however, it is never less than 255.

Note: This view is deprecated in favor of the ALL_PLSQL_OBJECT_SETTINGS view. Oracle recommends that you use ALL_PLSQL_OBJECT_SETTINGS instead. ALL_STORED_SETTINGS is retained for backward compatibility only.

ALL_STREAMS_GLOBAL_RULES

ALL_STREAMS_GLOBAL_RULES displays information about the following types of rules:

- Global rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Global rules created for the Streams propagations that have a source queue accessible to the current user
- Global rules created for the Streams apply processes that dequeue events from queues accessible to the current user

This view only contains information about rules created using the ADD_GLOBAL_RULES or ADD_GLOBAL_PROPAGATION_RULES procedures in the DBMS_STREAMS_ADM package. It does not contain information about rules created using the DBMS_RULE_ADM package.

Related View

DBA_STREAMS_GLOBAL_RULES displays information about the global rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process or propagation: <ul style="list-style-type: none"> ■ CAPTURE ■ PROPAGATION ■ APPLY
RULE_TYPE	VARCHAR2(7)		Type of rule (DML or DDL)
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-null tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)		Name of the rule
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: ["DBA_STREAMS_GLOBAL_RULES"](#) on page 3-126

ALL_STREAMS_MESSAGE_CONSUMERS

ALL_STREAMS_MESSAGE_CONSUMERS displays information about the Streams messaging clients accessible to the current user.

Related View

DBA_STREAMS_MESSAGE_CONSUMERS displays information about all Streams messaging clients in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)	NOT NULL	Name of the messaging client
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set
NOTIFICATION_TYPE	VARCHAR2(9)		Type of the notification action: <ul style="list-style-type: none"> ▪ PROCEDURE ▪ MAIL ▪ HTTP
NOTIFICATION_ACTION	VARCHAR2(256)		Notification action
NOTIFICATION_CONTEXT	ANYDATA		Context for the notification action

See Also: ["DBA_STREAMS_MESSAGE_CONSUMERS"](#) on page 3-126

ALL_STREAMS_MESSAGE_RULES

ALL_STREAMS_MESSAGE_RULES displays information about the Streams messaging rules accessible to the current user.

Related View

DBA_STREAMS_MESSAGE_RULES displays information about all Streams messaging rules in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)	NOT NULL	Name of the Streams client
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams client: <ul style="list-style-type: none"> ▪ PROPAGATION - Streams propagation ▪ APPLY - Streams apply process ▪ DEQUEUE - Streams messaging client
MESSAGE_TYPE_NAME	VARCHAR2(30)		Name of the message type
MESSAGE_TYPE_OWNER	VARCHAR2(30)		Owner of the message type
MESSAGE_RULE_VARIABLE	VARCHAR2(30)		Name of the variable in the message rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		Rule condition

See Also: ["DBA_STREAMS_MESSAGE_RULES"](#) on page 3-126

ALL_STREAMS_NEWLY_SUPPORTED

ALL_STREAMS_NEWLY_SUPPORTED displays information about the tables accessible to the current user that are newly supported by Streams.

Related View

DBA_NEWLY_SUPPORTED displays information about all tables in the database that are newly supported by Streams.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
REASON	VARCHAR2(32)		Reason why the table was not supported in a previous release: <ul style="list-style-type: none"> ▪ IOT ▪ column with user-defined type ▪ unsupported column exists ▪ object table ▪ AQ queue table ▪ temporary table ▪ sub object ▪ external table ▪ materialized view ▪ FILE column exists ▪ materialized view log ▪ materialized view container table ▪ streams unsupported object ▪ domain index
COMPATIBLE	CHAR(4)		The latest COMPATIBLE setting when the table was unsupported

See Also: ["DBA_STREAMS_NEWLY_SUPPORTED"](#) on page 3-126

ALL_STREAMS_RULES

ALL_STREAMS_RULES displays information about the rules used by the Streams processes accessible to the current user.

Related View

DBA_STREAMS_RULES displays information about the rules used by all Streams processes in the database.

Column	Datatype	NULL	Description
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process: <ul style="list-style-type: none"> ▪ CAPTURE ▪ PROPAGATION ▪ APPLY ▪ DEQUEUE
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set
RULE_SET_NAME	VARCHAR2(30)		Name of the rule set
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_CONDITION	CLOB		Current rule condition

Column	Datatype	NULL	Description
RULE_SET_TYPE	CHAR(8)		Type of the rule set: <ul style="list-style-type: none"> ■ POSITIVE ■ NEGATIVE
STREAMS_RULE_TYPE	VARCHAR2(6)		For global, schema, or table rules, the type of the rule: <ul style="list-style-type: none"> ■ TABLE ■ SCHEMA ■ GLOBAL
SCHEMA_NAME	VARCHAR2(30)		For table and schema rules, the schema name
OBJECT_NAME	VARCHAR2(30)		For table rules, the table name
SUBSETTING_OPERATION	VARCHAR2(6)		For subset rules, the type of operation: <ul style="list-style-type: none"> ■ INSERT ■ UPDATE ■ DELETE
DML_CONDITION	VARCHAR2(4000)		For subset rules, the row subsetting condition
INCLUDE_TAGGED_LCR	VARCHAR2(3)		For global, schema or table rules, indicates whether to include tagged LCRs (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		For global, schema, or table rules, the name of the database where the LCRs originated
RULE_TYPE	VARCHAR2(3)		For global, schema, or table rules, the type of rule: <ul style="list-style-type: none"> ■ DML ■ DDL
MESSAGE_TYPE_OWNER	VARCHAR2(30)		For message rules, the owner of the message type
MESSAGE_TYPE_NAME	VARCHAR2(30)		For message rules, the name of the message type
MESSAGE_RULE_VARIABLE	VARCHAR2(30)		For message rules, the name of the variable in the message rule
ORIGINAL_RULE_CONDITION	VARCHAR2(4000)		For rules created by Streams administrative APIs, the original rule condition when the rule was created
SAME_RULE_CONDITION	VARCHAR2(3)		For rules created by Streams administrative APIs, indicates whether the current rule condition is the same as the original rule condition (YES) or not (NO)

See Also: ["DBA_STREAMS_RULES"](#) on page 3-126

ALL_STREAMS_SCHEMA_RULES

ALL_STREAMS_SCHEMA_RULES displays information about the following types of rules:

- Schema rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Schema rules created for the Streams propagations that have a source queue accessible to the current user
- Schema rules created for the Streams apply processes that dequeue events from queues accessible to the current user

This view only contains information about rules created using the ADD_SCHEMA_RULES or ADD_SCHEMA_PROPAGATION_RULES procedures in the DBMS_STREAMS_ADM package. It does not contain information about rules created using the DBMS_RULE_ADM package.

Related View

DBA_STREAMS_SCHEMA_RULES displays information about the schema rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process or propagation: <ul style="list-style-type: none"> ■ CAPTURE ■ PROPAGATION ■ APPLY
SCHEMA_NAME	VARCHAR2(30)		Schema name in the rule condition. The rule evaluates to <code>true</code> for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this schema name.
RULE_TYPE	VARCHAR2(7)		Type of rule (DML or DDL)
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-null tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to <code>true</code> for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)		Name of the rule
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: ["DBA_STREAMS_SCHEMA_RULES"](#) on page 3-126

ALL_STREAMS_TABLE_RULES

ALL_STREAMS_TABLE_RULES displays information about the following types of rules:

- Table rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Table rules created for the Streams propagations that have a source queue accessible to the current user
- Table rules created for the Streams apply processes that dequeue events from queues accessible to the current user
- Subset rules created for the Streams apply processes that have a source queue accessible to the current user

This view only contains information about rules created using the `ADD_TABLE_RULES`, `ADD_TABLE_PROPAGATION_RULES`, or `ADD_SUBSET_RULES` procedures in the `DBMS_STREAMS_ADM` package. It does not contain information about rules created using the `DBMS_RULE_ADM` package.

Related View

DBA_STREAMS_TABLE_RULES displays information about the table rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process or propagation: <ul style="list-style-type: none"> ▪ CAPTURE ▪ PROPAGATION ▪ APPLY
TABLE_OWNER	VARCHAR2(30)		Table owner in the rule condition. The rule evaluates to <code>true</code> for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this table owner.
TABLE_NAME	VARCHAR2(30)		Table name in the rule condition. The rule evaluates to <code>true</code> for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this table name.
RULE_TYPE	VARCHAR2(7)		Type of rule (DML or DDL)
DML_CONDITION	VARCHAR2(4000)		Row subsetting condition, if the rule is a subset rule
SUBSETTING_OPERATION	VARCHAR2(6)		DML operation for row subsetting in the rule condition, if the rule is a subset rule: <ul style="list-style-type: none"> ▪ INSERT ▪ UPDATE ▪ DELETE <p>The rule evaluates to <code>true</code> for a logical change record (LCR) only if the LCR contains this command type after internal transformation.</p>
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-null tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to <code>true</code> for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)		Name of the rule
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: ["DBA_STREAMS_TABLE_RULES"](#) on page 3-126

ALL_STREAMS_TRANSFORM_FUNCTION

ALL_STREAMS_TRANSFORM_FUNCTION displays information about the rule-based transformation functions accessible to the current user.

Related View

DBA_STREAMS_TRANSFORM_FUNCTION displays information about all rule-based transformation functions in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule associated with the transformation function
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule associated with the transformation function
VALUE_TYPE	VARCHAR2(4000)		Type of the transformation function name. This type must be VARCHAR2 for a rule-based transformation to work properly.

Column	Datatype	NULL	Description
TRANSFORM_FUNCTION_NAME	VARCHAR2(4000)		Name of the transformation function (null if VALUE_TYPE is not VARCHAR2)

See Also: ["DBA_STREAMS_TRANSFORM_FUNCTION"](#) on page 3-127

ALL_STREAMS_UNSUPPORTED

ALL_STREAMS_UNSUPPORTED displays information about the tables accessible to the current user that are not supported by Streams in this release of the Oracle Database.

Related View

DBA_STREAMS_UNSUPPORTED displays information about all tables in the database that are not supported by Streams in this release of the Oracle Database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
REASON	VARCHAR2(39)		Reason why the table is not supported: <ul style="list-style-type: none"> ▪ IOT ▪ column with user-defined type ▪ unsupported column exists ▪ object table ▪ AQ queue table ▪ temporary table ▪ sub object ▪ external table ▪ materialized view ▪ FILE column exists ▪ materialized view log ▪ materialized view container table ▪ streams unsupported object ▪ domain index ▪ IOT with overflow ▪ IOT with LOB ▪ IOT with physical Rowid mapping ▪ mapping table for physical rowid of IOT ▪ IOT with LOB ▪ IOT with row movement ▪ summary container table
AUTO_FILTERED	VARCHAR2(3)		Indicates whether Streams automatically filters out the object (YES) or not (NO)

See Also: ["DBA_STREAMS_UNSUPPORTED"](#) on page 3-127

ALL_SUBPART_COL_STATISTICS

ALL_SUBPART_COL_STATISTICS describes column statistics and histogram information for subpartitions of partitioned objects accessible to the current user.

Related Views

- `DBA_SUBPART_COL_STATISTICS` provides this information for all subpartitions in the database.
- `USER_SUBPART_COL_STATISTICS` provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner name
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Table name
<code>SUBPARTITION_NAME</code>	<code>VARCHAR2(30)</code>		Table subpartition name
<code>COLUMN_NAME</code>	<code>VARCHAR2(4000)</code>		Column name
<code>NUM_DISTINCT</code>	NUMBER		Number of distinct values in the column
<code>LOW_VALUE</code>	<code>RAW(32)</code>		Low value in the column
<code>HIGH_VALUE</code>	<code>RAW(32)</code>		High value in the column
<code>DENSITY</code>	NUMBER		Density of the column
<code>NUM_NULLS</code>	NUMBER		Number of nulls in the column
<code>NUM_BUCKETS</code>	NUMBER		Number of buckets in histogram for the column
<code>SAMPLE_SIZE</code>	NUMBER		Sample size used in analyzing this column
<code>LAST_ANALYZED</code>	DATE		Date on which this column was most recently analyzed
<code>GLOBAL_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO)
<code>USER_STATS</code>	<code>VARCHAR2(3)</code>		Indicates whether statistics were entered directly by the user (YES) or not (NO)
<code>AVG_COL_LEN</code>	NUMBER		Average length of the column (in bytes)
<code>HISTOGRAM</code>	<code>VARCHAR2(15)</code>		Indicates existence/type of histogram: <ul style="list-style-type: none"> ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

ALL_SUBPART_HISTOGRAMS

`ALL_SUBPART_HISTOGRAMS` lists the actual histogram data (end-points per histogram) for histograms on table subpartitions accessible to the current user.

Related Views

- `DBA_SUBPART_HISTOGRAMS` provides this information for all subpartitions in the database.
- `USER_SUBPART_HISTOGRAMS` provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the `OWNER` column.

Note: These views are populated only if you called statistics on the index using the `ANALYZE` statement or the `DBMS_STATS` package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner name
TABLE_NAME	VARCHAR2(30)		Table name
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket

ALL_SUBPART_KEY_COLUMNS

ALL_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) accessible to the current user.

Related Views

- DBA_SUBPART_KEY_COLUMNS provides this information for all subpartitions in the database.
- USER_SUBPART_KEY_COLUMNS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		The partitioned table or index owner
NAME	VARCHAR2(30)		The partitioned table or index name
OBJECT_TYPE	VARCHAR2(11)		The object type (TABLE or INDEX)
COLUMN_NAME	VARCHAR2(30)		The column name
COLUMN_POSITION	NUMBER		The position of the column within the subpartitioning key

See Also:

- ["DBA_SUBPART_KEY_COLUMNS"](#) on page 3-127
- ["USER_SUBPART_KEY_COLUMNS"](#) on page 3-181

ALL_SUBPARTITION_TEMPLATES

ALL_SUBPARTITION_TEMPLATES describes the subpartition templates accessible to the current user.

Related Views

- DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database.
- USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)	NOT NULL	Owner of the table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
SUBPARTITION_NAME	VARCHAR2(34)	NOT NULL	Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name of the subpartition
HIGH_BOUND	LONG		Literal list values of the subpartition

See Also:

- ["DBA_SUBPARTITION_TEMPLATES"](#) on page 3-127
- ["USER_SUBPARTITION_TEMPLATES"](#) on page 3-181

ALL_SUBSCRIBED_COLUMNS

ALL_SUBSCRIBED_COLUMNS describes the columns of source tables to which any subscription accessible to the current user has subscribed. This view is intended for use by Change Data Capture subscribers.

Related Views

- DBA_SUBSCRIBED_COLUMNS describes the columns of source tables to which any subscriber has subscribed.
- USER_SUBSCRIBED_COLUMNS describes the columns of source tables to which the current user has subscribed.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Source table schema identifier
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Source table identifier
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Source table column identifier
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription

See Also:

- ["DBA_SUBSCRIBED_COLUMNS"](#) on page 3-127
- ["USER_SUBSCRIBED_COLUMNS"](#) on page 3-181

ALL_SUBSCRIBED_TABLES

ALL_SUBSCRIBED_TABLES describes the source tables to which any subscription accessible to the current user has subscribed. This view is intended for use by Change Data Capture subscribers.

Related Views

- DBA_SUBSCRIBED_TABLES describes all source tables in the database to which any subscriber has subscribed.
- USER_SUBSCRIBED_TABLES describes the source tables to which the current user has subscribed.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Source table schema identifier
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Source table identifier
VIEW_NAME	VARCHAR2(30)		Subscriber view name
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Change set identifier
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription

See Also:

- ["DBA_SUBSCRIBED_TABLES"](#) on page 3-128
- ["USER_SUBSCRIBED_TABLES"](#) on page 3-181

ALL_SUBSCRIPTIONS

ALL_SUBSCRIPTIONS describes the subscriptions accessible to the current user. This view is intended for use by Change Data Capture subscribers.

Related Views

- DBA_SUBSCRIPTIONS describes all subscriptions in the database.
- USER_SUBSCRIPTIONS describes the subscriptions owned by the current user.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handler
SET_NAME	VARCHAR2(30)	NOT NULL	Change set identifier
USERNAME	VARCHAR2(30)	NOT NULL	User name of the subscriber
CREATED	DATE	NOT NULL	Subscription creation date
STATUS	VARCHAR2(1)	NOT NULL	N indicates not yet active, A indicates is currently active
EARLIEST_SCN	NUMBER	NOT NULL	Starting point for the select/view
LATEST_SCN	NUMBER	NOT NULL	Ending point for the select/view
DESCRIPTION	VARCHAR2(255)		Comment field for the subscriber
LAST_PURGED	DATE		Last time the subscriber called the PURGE_WINDOW for this subscription
LAST_EXTENDED	DATE		Last time the subscriber called the EXTEND_WINDOW for this subscription
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription

See Also:

- ["DBA_SUBSCRIPTIONS"](#) on page 3-128
- ["USER_SUBSCRIPTIONS"](#) on page 3-182

ALL_SUMDELTA

ALL_SUMDELTA lists direct path load entries accessible to the current user.

Column	Datatype	NULL	Description
TABLEOBJ#	NUMBER	NOT NULL	Object number of the table

Column	Datatype	NULL	Description
PARTITIONOBJ#	NUMBER	NOT NULL	Object number of table partitions (if the table is partitioned)
DMLOPERATION	VARCHAR2(1)		Type of DML operation applied to the table
SCN	NUMBER	NOT NULL	SCN when the bulk DML occurred
TIMESTAMP	DATE	NOT NULL	Timestamp of log entry
LOWROWID	ROWID	NOT NULL	The start ROWID in the loaded rowid range
HIGHROWID	ROWID	NOT NULL	The end ROWID in the loaded rowid range

ALL_SYNONYMS

ALL_SYNONYMS describes the synonyms accessible to the current user.

Related Views

- DBA_SYNONYMS describes all synonyms in the database.
- USER_SYNONYMS describes the synonyms owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym
SYNONYM_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym
TABLE_OWNER	VARCHAR2(30)		Owner of the object referenced by the synonym
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object referenced by the synonym
DB_LINK	VARCHAR2(128)		Name of the database link referenced, if any

See Also:

- ["DBA_SYNONYMS"](#) on page 3-128
- ["USER_SYNONYMS"](#) on page 3-182

ALL_TAB_COL_STATISTICS

ALL_TAB_COL_STATISTICS provides column statistics and histogram information extracted from ["ALL_TAB_COLUMNS"](#) on page 2-124.

Related Views

- DBA_TAB_COL_STATISTICS provides such information extracted from ["DBA_TAB_COLUMNS"](#) on page 3-129.
- USER_TAB_COL_STATISTICS provides such information extracted from ["USER_TAB_COLUMNS"](#) on page 3-182.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Table owner
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column

Column	Datatype	NULL	Description
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: <ul style="list-style-type: none"> ▪ NONE ▪ FREQUENCY ▪ HEIGHT BALANCED

ALL_TAB_COLS

ALL_TAB_COLS describes the columns of the tables, views, and clusters accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

This view differs from "ALL_TAB_COLUMNS" on page 2-124 in that hidden columns are not filtered out.

Related Views

- DBA_TAB_COLS describes the columns of all tables, views, and clusters in the database.
- USER_TAB_COLS describes the columns of the tables, views, and clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table, view, or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table, view, or cluster
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER		Sequence number of the column as created

Column	Datatype	NULL	Description
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column ¹
LOW_VALUE	RAW (32)		Low value in the column ¹
HIGH_VALUE	RAW (32)		High value in the column ¹
DENSITY	NUMBER		Density of the column ¹
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column Note: The number of buckets in a histogram is specified in the <i>SIZE</i> parameter of the SQL statement <i>ANALYZE</i> . However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2 (44)		Name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2 (3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2 (3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2 (1)		B C. B indicates that the column uses BYTE length semantics. C indicates that the column uses CHAR length semantics. NULL indicates the datatype is not any of the following: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2 (3)		Indicates whether the column data is in release 8.0 image format (YES) or not (NO)
DATA_UPGRADED	VARCHAR2 (3)		Indicates whether the column data has been upgraded to the latest type version format (YES) or not (NO)
HIDDEN_COLUMN	VARCHAR2 (3)		Indicates whether the column is a hidden column (YES) or not (NO)
VIRTUAL_COLUMN	VARCHAR2 (3)		Indicates whether the column is a virtual column (YES) or not (NO)
SEGMENT_COLUMN_ID	NUMBER		Sequence number of the column in the segment
INTERNAL_COLUMN_ID	NUMBER	NOT NULL	Internal sequence number of the column

Column	Datatype	NULL	Description
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: <ul style="list-style-type: none"> ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED
QUALIFIED_COL_NAME	VARCHAR2(4000)		Qualified column name

¹ These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views.

ALL_TAB_COLUMNS

ALL_TAB_COLUMNS describes the columns of the tables, views, and clusters accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

Related Views

- DBA_TAB_COLUMNS describes the columns of all tables, views, and clusters in the database.
- USER_TAB_COLUMNS describes the columns of the tables, views, and clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table, view, or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table, view, or cluster
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER		Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column ¹
LOW_VALUE	RAW(32)		Low value in the column ¹
HIGH_VALUE	RAW(32)		High value in the column ¹
DENSITY	NUMBER		Density of the column ¹
NUM_NULLS	NUMBER		Number of nulls in the column

Column	Datatype	NULL	Description
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column Note: The number of buckets in a histogram is specified in the <code>SIZE</code> parameter of the SQL statement <code>ANALYZE</code> . However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set: <code>CHAR_CS</code> or <code>NCHAR_CS</code>
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (<code>YES</code>) or were estimated from statistics on underlying partitions and subpartitions (<code>NO</code>)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (<code>YES</code>) or not (<code>NO</code>)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2(1)		B C. B indicates that the column uses <code>BYTE</code> length semantics. C indicates that the column uses <code>CHAR</code> length semantics. NULL indicates the datatype is not any of the following: <ul style="list-style-type: none"> ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2(3)		Indicates whether the column data is in release 8.0 image format (<code>YES</code>) or not (<code>NO</code>)
DATA_UPGRADED	VARCHAR2(3)		Indicates whether the column data has been upgraded to the latest type version format (<code>YES</code>) or not (<code>NO</code>)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: <ul style="list-style-type: none"> ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

¹ These columns remain for backward compatibility with Oracle7. This information is now in the `{TAB | PART}_COL_STATISTICS` views.

ALL_TAB_COMMENTS

`ALL_TAB_COMMENTS` displays comments on the tables and views accessible to the current user.

Related Views

- `DBA_TAB_COMMENTS` displays comments on all tables and views in the database.

- `USER_TAB_COMMENTS` displays comments on the tables and views owned by the current user. This view does not display the `OWNER` column.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>	NOT NULL	Owner of the object
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>	NOT NULL	Name of the object
<code>TABLE_TYPE</code>	<code>VARCHAR2(11)</code>		Type of the object
<code>COMMENTS</code>	<code>VARCHAR2(4000)</code>		Comment on the object

See Also:

- ["DBA_TAB_COMMENTS"](#) on page 3-129
- ["USER_TAB_COMMENTS"](#) on page 3-182

ALL_TAB_HISTOGRAMS

`ALL_TAB_HISTOGRAMS` describes histograms on tables and views accessible to the current user.

Related Views

- `DBA_TAB_HISTOGRAMS` describes histograms on all tables and views in the database.
- `USER_TAB_HISTOGRAMS` describes histograms on all tables and views owned by the current user. This view does not display the `OWNER` column.

Note: These views are populated only if you called statistics on the index using the `ANALYZE` statement or the `DBMS_STATS` package.

Column	Datatype	NULL	Description
<code>OWNER</code>	<code>VARCHAR2(30)</code>		Owner of table
<code>TABLE_NAME</code>	<code>VARCHAR2(30)</code>		Name of the table
<code>COLUMN_NAME</code>	<code>VARCHAR2(4000)</code>		Column name or attribute of the object type column
<code>ENDPOINT_NUMBER</code>	<code>NUMBER</code>		Histogram bucket number
<code>ENDPOINT_VALUE</code>	<code>NUMBER</code>		Normalized endpoint value for this bucket
<code>ENDPOINT_ACTUAL_VALUE</code>	<code>VARCHAR2(1000)</code>		Actual (not normalized) string value of the endpoint for this bucket

ALL_TAB_MODIFICATIONS

`ALL_TAB_MODIFICATIONS` describes tables accessible to the current user that have been modified since the last time statistics were gathered on the tables.

Related Views

- `DBA_TAB_MODIFICATIONS` provides such information for all tables in the database.
- `USER_TAB_MODIFICATIONS` provides such information for tables owned by the current user. This view does not display the `TABLE_OWNER` column.

Note: These views are populated only for tables with the MONITORING attribute. They are intended for statistics collection over a long period of time. For performance reasons, the Oracle Database may not populate these views until a few hours after the actual modifications occurred.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the modified table.
TABLE_NAME	VARCHAR2(30)		Name of the modified table
PARTITION_NAME	VARCHAR2(30)		Name of the modified partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the modified subpartition
INSERTS	NUMBER		Approximate number of inserts since the last time statistics were gathered
UPDATES	NUMBER		Approximate number of updates since the last time statistics were gathered
DELETES	NUMBER		Approximate number of deletes since the last time statistics were gathered
TIMESTAMP	DATE		Indicates the last time the table was modified
DROP_SEGMENTS	NUMBER		Number of partition and subpartition segments dropped since the last analyze

ALL_TAB_PARTITIONS

ALL_TAB_PARTITIONS provides partition-level partitioning information, partition storage parameters, and partition statistics collected by ANALYZE statements for partitions accessible to the current user.

Related Views

- DBA_TAB_PARTITIONS provides such information for all partitions in the database.
- USER_TAB_PARTITIONS provides such information for partitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COMPOSITE	VARCHAR2(3)		YES if the partition belongs to a Local index on a table partitioned using Composite method; NO otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block

Column	Datatype	NULL	Description
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
NUM_ROWS	NUMBER		Number of rows in the partition
BLOCKS	NUMBER		Number of used blocks in the partition
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the partition
AVG_SPACE	NUMBER		Average available free space in the partition
CHAIN_CNT	NUMBER		Number of chained rows in the partition
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool to be used for the partition blocks
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

ALL_TAB_PRIVS

ALL_TAB_PRIVS describes the following types of grants:

- Object grants for which the current user is the object owner, grantor, or grantee
- Object grants for which an enabled role or PUBLIC is the grantee

Related Views

- DBA_TAB_PRIVS describes all object grants in the database.
- USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

Column	Datatype	NULL	Description
HIERARCHY	VARCHAR2 (3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also:

- ["DBA_TAB_PRIVS"](#) on page 3-129
- ["USER_TAB_PRIVS"](#) on page 3-183

ALL_TAB_PRIVS_MADE

ALL_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner or grantor.

Related View

USER_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2 (30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2 (30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2 (40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2 (3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2 (3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: ["USER_TAB_PRIVS_MADE"](#) on page 3-183

ALL_TAB_PRIVS_RECD

ALL_TAB_PRIVS_RECD describes the following types of grants:

- Object grants for which the current user is the grantee
- Object grants for which an enabled role or PUBLIC is the grantee

Related View

USER_TAB_PRIVS_RECD describes the object grants for which the current user is the grantee. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2 (30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2 (30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2 (40)	NOT NULL	Privilege on the object

Column	Datatype	NULL	Description
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: ["USER_TAB_PRIVS_REC'D"](#) on page 3-183

ALL_TAB_STATISTICS

ALL_TAB_STATISTICS displays optimizer statistics for the tables accessible to the current user.

Related Views

- DBA_TAB_STATISTICS displays optimizer statistics for all tables in the database.
- USER_TAB_STATISTICS displays optimizer statistics for the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
PARTITION_POSITION	NUMBER		Position of the partition within the table
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition within the partition
OBJECT_TYPE	VARCHAR2(12)		Type of the object: <ul style="list-style-type: none"> ■ TABLE ■ PARTITION ■ SUBPARTITION
NUM_ROWS	NUMBER		Number of rows in the object
BLOCKS	NUMBER		Number of used blocks in the object
EMPTY_BLOCKS	NUMBER		Number of empty blocks in the object
AVG_SPACE	NUMBER		Average available free space in the object
CHAIN_CNT	NUMBER		Number of chained rows in the object
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
AVG_CACHED_BLOCKS	NUMBER		Average number of blocks in the buffer cache
AVG_CACHE_HIT_RATIO	NUMBER		Average cache hit ratio for the object
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the table
LAST_ANALYZED	DATE		Date of the most recent time the table was analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were calculated without merging underlying partitions (YES) or not (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

Column	Datatype	NULL	Description
STATTYPE_LOCKED	VARCHAR2 (5)		Type of statistics lock: <ul style="list-style-type: none"> ■ DATA ■ CACHE ■ ALL

See Also:

- ["DBA_TAB_STATISTICS"](#) on page 3-130
- ["USER_TAB_STATISTICS"](#) on page 3-183

ALL_TAB_SUBPARTITIONS

ALL_TAB_SUBPARTITIONS describes, for each table subpartition accessible to the current user, the subpartition name, name of the table and partition to which it belongs, and its storage attributes.

Related Views

- DBA_TAB_SUBPARTITIONS provides such information for all subpartitions in the database.
- USER_TAB_SUBPARTITIONS provides such information for subpartitions of all partitioned objects owned by the current user. This view does not display the TABLE_OWNER column.

Note: Statistics are not collected on a per-subpartition basis.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2 (30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2 (30)		Partition name
SUBPARTITION_NAME	VARCHAR2 (30)		Subpartition name
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
TABLESPACE_NAME	VARCHAR2 (30)	NOT NULL	Name of the tablespace containing the subpartition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of freelist groups allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2 (3)		Logging attribute of subpartition
NUM_ROWS	NUMBER		The number of rows

Column	Datatype	NULL	Description
BLOCKS	NUMBER		The number of blocks
EMPTY_BLOCKS	NUMBER		The number of empty blocks
AVG_SPACE	NUMBER		The average space
CHAIN_CNT	NUMBER		The chain count
AVG_ROW_LEN	NUMBER		The average row length
SAMPLE_SIZE	NUMBER		The sample size
LAST_ANALYZED	DATE		The date on which this table was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics collected for partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

ALL_TABLES

ALL_TABLES describes the relational tables accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement.

Related Views

- DBA_TABLES describes all relational tables in the database.
- USER_TABLES describes the relational tables owned by the current user. This view does not display the OWNER column.

Note: Columns marked with an asterisk (*) are populated only if you collect statistics on the table with the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; null for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
PCT_FREE	NUMBER		Minimum percentage of free space in a block; null for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; null for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; null for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; null for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); null for partitioned tables

Column	Datatype	NULL	Description
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); null for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; null for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; null for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; null for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; null for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment; null for partitioned tables
LOGGING	VARCHAR2 (3)		Logging attribute; NULL for partitioned tables
BACKED_UP	VARCHAR2 (1)		Has table been backed up since last change
NUM_ROWS*	NUMBER		Number of rows in the table
BLOCKS*	NUMBER		Number of used data blocks in the table
EMPTY_BLOCKS*	NUMBER		Number of empty (never used) data blocks in the table
AVG_SPACE*	NUMBER		Average amount of free space, in bytes, in a data block allocated to the table
CHAIN_CNT*	NUMBER		Number of rows in the table that are chained from one data block to another, or which have migrated to a new block, requiring a link to preserve the old ROWID
AVG_ROW_LEN*	NUMBER		Average length of a row in the table (in bytes)
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2 (10)		Number of threads per instance for scanning the table
INSTANCES	VARCHAR2 (10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2 (5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)
TABLE_LOCK	VARCHAR2 (8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2 (3)		Indicates whether this table is partitioned. Set to YES if it is partitioned.
IOT_TYPE	VARCHAR2 (12)		If this is an index-organized table, then IOT_TYPE is IOT, IOT_OVERFLOW, or IOT_MAPPING. If this is not an index-organized table, then IOT_TYPE is NULL.
TEMPORARY	VARCHAR2 (1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2 (1)		Whether the trigger is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y N)
NESTED	VARCHAR2 (3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2 (7)		The default buffer pool for the object. NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2 (8)		Whether partitioned row movement is enabled or disabled

Column	Datatype	NULL	Description
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: <ul style="list-style-type: none"> ■ SYS\$SESSION: the rows are preserved for the duration of the session ■ SYS\$TRANSACTION: the rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2(8)		Whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set
CLUSTER_OWNER	VARCHAR2(30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2(8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); null for partitioned tables
DROPPED	VARCHAR2(3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); null for partitioned tables

See Also:

- ["DBA_TABLES"](#) on page 3-130
- ["USER_TABLES"](#) on page 3-184

ALL_TRIGGER_COLS

ALL_TRIGGER_COLS describes the use of columns in triggers owned by user or in triggers on tables owned by user. If the user has the CREATE ANY TRIGGER privilege, this view describes the use of columns in all triggers.

Related Views

- DBA_TRIGGER_COLS provides such information for all triggers in the database.
- USER_TRIGGER_COLS provides such information for all triggers owned by the current user.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)		Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)		Name of the trigger
TABLE_OWNER	VARCHAR2(30)		Owner of the table on which the trigger is defined
TABLE_NAME	VARCHAR2(30)		Table on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column used in the trigger
COLUMN_LIST	VARCHAR2(3)		Column specified in UPDATE clause (Y N)

Column	Datatype	NULL	Description
COLUMN_USAGE	VARCHAR2(17)		How the column is used in the trigger. All applicable combinations of NEW, OLD, IN, OUT, and IN OUT.

ALL_TRIGGERS

ALL_TRIGGERS describes owned by the current user and triggers on tables owned by the current user. If the user has the CREATE ANY TRIGGER privilege, then this view describes all triggers in the database.

Related Views

- DBA_TRIGGERS describes all triggers in the database.
- USER_TRIGGERS describes all triggers owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)		Name of the trigger
TRIGGER_TYPE	VARCHAR2(16)		When the trigger fires: BEFORE STATEMENT, BEFORE EACH ROW, BEFORE EVENT, AFTER STATEMENT, AFTER EACH ROW, and AFTER EVENT
TRIGGERING_EVENT	VARCHAR2(227)		The DML, DDL, or database event that fires the trigger. For a listing of triggering events, see the CREATE TRIGGER statement in <i>Oracle Database SQL Reference</i> .
TABLE_OWNER	VARCHAR2(30)		Owner of the table on which the trigger is defined
BASE_OBJECT_TYPE	VARCHAR2(16)		Base object on which the trigger is defined: TABLE, VIEW, SCHEMA, or DATABASE
TABLE_NAME	VARCHAR2(30)		If the base object type of the trigger is SCHEMA or DATABASE, then this column is NULL; if the base object type of the trigger is TABLE or VIEW, this column indicates the table/view name on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the nested table column (if nested table trigger), else null
REFERENCING_NAMES	VARCHAR2(128)		Names used for referencing OLD and NEW column values from within the trigger
WHEN_CLAUSE	VARCHAR2(4000)		Must evaluate to TRUE for TRIGGER_BODY to execute
STATUS	VARCHAR2(8)		Whether the trigger is enabled (ENABLED DISABLED)
DESCRIPTION	VARCHAR2(4000)		Trigger description; useful for re-creating a trigger creation statement
ACTION_TYPE	VARCHAR2(11)		Action type of the trigger body (CALL or PL/SQL)
TRIGGER_BODY	LONG		Statements executed by the trigger when it fires

ALL_TYPE_ATTRS

ALL_TYPE_ATTRS describes the attributes of the object types accessible to the current user.

Related Views

- DBA_TYPE_ATTRS describes the attributes of all object types in the database.
- USER_TYPE_ATTRS describes the attributes of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)		Name of the type
ATTR_NAME	VARCHAR2(30)		Name of the attribute
ATTR_TYPE_MOD	VARCHAR2(7)		Type modifier of the attribute: <ul style="list-style-type: none"> ▪ REF ▪ POINTER
ATTR_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the attribute
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute, or maximum length of the VARCHAR or VARCHAR2 attribute.
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute, or binary precision of the FLOAT attribute.
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET_NAME	VARCHAR2(44)		Character set name of the attribute (CHAR_CS or NCHAR_CS)
ATTR_NO	NUMBER		Syntactical order number or position of the attribute as specified in the type specification or CREATE TYPE statement (not to be used as an ID number)
INHERITED	VARCHAR2(3)		Indicates whether the attribute is inherited from a supertype (YES) or not (NO)

See Also:

- ["DBA_TYPE_ATTRS"](#) on page 3-134
- ["USER_TYPE_ATTRS"](#) on page 3-185

ALL_TYPE_METHODS

ALL_TYPE_METHODS describes the methods of the object types accessible to the current user.

Related Views

- DBA_TYPE_METHODS describes the methods of all object types in the database.
- USER_TYPE_METHODS describes the methods of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	Method number for distinguishing overloaded method (not to be used as ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method
FINAL	VARCHAR2(3)		Indicates whether the method is final (YES) or not (NO)
INSTANTIABLE	VARCHAR2(3)		Indicates whether the method is instantiable (YES) or not (NO)

Column	Datatype	NULL	Description
OVERRIDING	VARCHAR2 (3)		Indicates whether the method is overriding a supertype method (YES) or not (NO)
INHERITED	VARCHAR2 (3)		Indicates whether the method is inherited from a supertype (YES) or not (NO)

See Also:

- ["DBA_TYPE_METHODS"](#) on page 3-134
- ["USER_TYPE_METHODS"](#) on page 3-185

ALL_TYPE_VERSIONS

ALL_TYPE_VERSIONS describes the versions of the object types accessible to the current user.

Related Views

- DBA_TYPE_VERSIONS describes the versions of all object types in the database.
- USER_TYPE_VERSIONS describes the versions of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2 (30)	NOT NULL	Name of the type
VERSION#	NUMBER	NOT NULL	Internal version number of the type
TYPECODE	VARCHAR2 (30)		Typecode of the type
STATUS	VARCHAR2 (7)		Status of the type: <ul style="list-style-type: none"> ■ N/A ■ VALID ■ INVALID
LINE	NUMBER	NOT NULL	Line number of the type's spec
TEXT	VARCHAR2 (4000)		Text of the type's spec
HASHCODE	RAW (17)		Hashcode of the type

See Also:

- ["DBA_TYPE_VERSIONS"](#) on page 3-134
- ["USER_TYPE_VERSIONS"](#) on page 3-185

ALL_TYPES

ALL_TYPES describes the object types accessible to the current user.

Related Views

- DBA_TYPES describes all object types in the database.
- USER_TYPES describes the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
TYPE_OID	RAW(16)	NOT NULL	Object identifier (OID) of the type
TYPECODE	VARCHAR2(30)		Typecode of the type
ATTRIBUTES	NUMBER		Number of attributes (if any) in the type
METHODS	NUMBER		Number of methods (if any) in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type (YES) or not (NO)
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type (YES) or not (NO)
FINAL	VARCHAR2(3)		Indicates whether the type is a final type (YES) or not (NO)
INSTANTIABLE	VARCHAR2(3)		Indicates whether the type is an instantiable type (YES) or not (NO)
SUPERTYPE_OWNER	VARCHAR2(30)		Owner of the supertype (NULL if type is not a subtype)
SUPERTYPE_NAME	VARCHAR2(30)		Name of the supertype (NULL if type is not a subtype)
LOCAL_ATTRIBUTES	NUMBER		Number of local (not inherited) attributes (if any) in the subtype
LOCAL_METHODS	NUMBER		Number of local (not inherited) methods (if any) in the subtype
TYPEID	RAW(16)		Type ID value of the type

See Also:

- ["DBA_TYPES"](#) on page 3-135
- ["USER_TYPES"](#) on page 3-185

ALL_UNUSED_COL_TABS

ALL_UNUSED_COL_TABS describes all tables accessible to the current user that contain unused columns.

Related Views

- DBA_UNUSED_COL_TABS describes all tables in the database that contain unused columns.
- USER_UNUSED_COL_TABS describes all tables owned by the current user that contain unused columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	The name of the table
COUNT	NUMBER		The number of unused columns

ALL_UPDATABLE_COLUMNS

ALL_UPDATABLE_COLUMNS describes all columns in a join view that are updatable by the current user.

Related Views

- `DBA_UPDATABLE_COLUMNS` describes all columns in a join view that are updatable by the database administrator.
- `USER_UPDATABLE_COLUMNS` describes all columns owned by the current user that are in a join view and are updatable by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

ALL_USERS

`ALL_USERS` lists all users of the database visible to the current user. This view does not describe the users (see the related views).

Related Views

- `DBA_USERS` describes all users of the database, and contains more columns than `ALL_USERS`.
- `USER_USERS` describes the current user, and contains more columns than `ALL_USERS`.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
CREATED	DATE	NOT NULL	User creation date

See Also:

- ["DBA_USERS"](#) on page 3-135
- ["USER_USERS"](#) on page 3-185

ALL_USTATS

`ALL_USTATS` describes all user-defined statistics accessible to the current user.

Related Views

- `DBA_USTATS` describes all user-defined statistics in the database.
- `USER_USTATS` describes all user-defined statistics owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or index for which the statistics have been collected
OBJECT_NAME	VARCHAR2(30)		Name of the table or index for which the statistics have been collected

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(6)		Indicates the type of object for which statistics have been collected: COLUMN or INDEX
ASSOCIATION	VARCHAR2(8)		DIRECT Indicates a direct association with the object for which the statistics have been collected. IMPLICIT indicates the association for which the statistics have been collected is with the column type or index type, and the object is an instance of that column type or index type.
COLUMN_NAME	VARCHAR2(30)		Column name, if property is column for which statistics have been collected
STATSTYPE_SCHEMA	VARCHAR2(30)		Schema of statistics type which was used to collect the statistics
STATSTYPE_NAME	VARCHAR2(30)		Name of statistics type which was used to collect statistics
STATISTICS	RAW(2000)		User collected statistics for the object
PARTITION_NAME	VARCHAR2(30)		Partition name of a table. If null, the table is either non-partitioned or the entry corresponds to the aggregate statistics for the table.

ALL_VARRAYS

ALL_VARRAYS describes all varrays accessible to the current user.

Related Views

- DBA_VARRAYS describes all varrays in the database.
- USER_VARRAYS describes all varrays owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table containing the varray
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the containing table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Name of the varray column or attribute
TYPE_OWNER	VARCHAR2(30)		Owner of the varray type
TYPE_NAME	VARCHAR2(30)		Name of the varray type
LOB_NAME	VARCHAR2(30)		Name of the LOB if the varray is stored in a LOB
STORAGE_SPEC	VARCHAR2(30)		DEFAULT value indicates that the storage was defaulted. USER_SPECIFIED value indicates that the storage was user-specified.
RETURN_TYPE	VARCHAR2(20)		Return type of the column
ELEMENT_SUBSTITUTABLE	VARCHAR2(25)		Indicates whether the varray element is substitutable (Y) or not (N)

ALL_VIEWS

ALL_VIEWS describes the views accessible to the current user.

Related Views

- DBA_VIEWS describes all views in the database.
- USER_VIEWS describes the views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the view
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the view
TEXT_LENGTH	NUMBER		Length of the view text
TEXT	LONG		View text
TYPE_TEXT_LENGTH	NUMBER		Length of the type clause of the typed view
TYPE_TEXT	VARCHAR2(4000)		Type clause of the typed view
OID_TEXT_LENGTH	NUMBER		Length of the WITH OID clause of the typed view
OID_TEXT	VARCHAR2(4000)		WITH OID clause of the typed view
VIEW_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the view if the view is a typed view
VIEW_TYPE	VARCHAR2(30)		Type of the view if the view is a typed view
SUPERVIEW_NAME	VARCHAR2(30)		Name of the superview

See Also:

- ["DBA_VIEWS"](#) on page 3-136
- ["USER_VIEWS"](#) on page 3-186

ALL_WARNING_SETTINGS

ALL_WARNING_SETTINGS displays information about the warning parameter settings for the objects accessible to the current user.

Related Views

- DBA_WARNING_SETTINGS displays information about the warning parameter settings for all objects in the database.
- USER_WARNING_SETTINGS displays information about the warning parameter settings for the objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
OBJECT_TYPE	VARCHAR2(12)		Type of the object: <ul style="list-style-type: none"> ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY
WARNING	VARCHAR2(40)		Warning number or category: <ul style="list-style-type: none"> ■ INFORMATIONAL ■ PERFORMANCE ■ SEVERE ■ ALL

Column	Datatype	NULL	Description
SETTING	VARCHAR2(7)		Value of the warning setting: <ul style="list-style-type: none"> ■ DISABLE ■ ENABLE ■ ERROR

See Also:

- ["DBA_WARNING_SETTINGS"](#) on page 3-137
- ["USER_WARNING_SETTINGS"](#) on page 3-186

ALL_XML_SCHEMAS

ALL_XML_SCHEMAS describes the registered XML schemas accessible to the current user.

Related Views

- DBA_XML_SCHEMAS describes all registered XML schemas in the database.
- USER_XML_SCHEMAS describes the registered XML schemas owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML schema
SCHEMA_URL	VARCHAR2(700)		Schema URL of the XML schema
LOCAL	VARCHAR2(3)		Indicates whether the XML schema is local (YES) or global (NO)
SCHEMA	XMLTYPE		XML schema document
INT_OBJNAME	VARCHAR2(4000)		Internal database object name for the schema
QUAL_SCHEMA_URL	VARCHAR2(767)		Fully qualified schema URL

See Also:

- ["DBA_XML_SCHEMAS"](#) on page 3-137
- ["USER_XML_SCHEMAS"](#) on page 3-186

ALL_XML_TAB_COLS

ALL_XML_TAB_COLS describes the columns of the XML tables accessible to the current user.

Related Views

- DBA_XML_TAB_COLS describes the columns of all XML tables in the database.
- USER_XML_TAB_COLS describes the columns of the XML tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML table
TABLE_NAME	VARCHAR2(30)		Name of the XML table

Column	Datatype	NULL	Description
COLUMN_NAME	VARCHAR2(4000)		Name of the XML table column
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2(256)		Name of the XML Schema element that is used for the table
STORAGE_TYPE	VARCHAR2(17)		Storage option for the XMLtype data (OBJECT-RELATIONAL or CLOB)

See Also:

- ["DBA_XML_TAB_COLS"](#) on page 3-137
- ["USER_XML_TAB_COLS"](#) on page 3-186

ALL_XML_TABLES

ALL_XML_TABLES describes the XML tables accessible to the current user.

Related Views

- DBA_XML_TABLES describes all XML tables in the database.
- USER_XML_TABLES describes the XML tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML table
TABLE_NAME	VARCHAR2(30)		Name of the XML table
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2(256)		Name of the XML Schema element that is used for the table
STORAGE_TYPE	VARCHAR2(17)		Storage option for the XMLtype data (OBJECT-RELATIONAL or CLOB)

See Also:

- ["DBA_XML_TABLES"](#) on page 3-137
- ["USER_XML_TABLES"](#) on page 3-186

ALL_XML_VIEW_COLS

ALL_XML_VIEW_COLS describes the columns of the XML views accessible to the current user.

Related Views

- DBA_XML_VIEW_COLS describes the columns of all XML views in the database.
- USER_XML_VIEW_COLS describes the columns of the XML views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML view
VIEW_NAME	VARCHAR2(30)		Name of the XML view
COLUMN_NAME	VARCHAR2(4000)		Name of the XML view column
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2(256)		Name of the XML Schema element that is used for the view

See Also:

- ["DBA_XML_VIEW_COLS"](#) on page 3-137
- ["USER_XML_VIEW_COLS"](#) on page 3-186

ALL_XML_VIEWS

ALL_XML_VIEWS describes the XML views accessible to the current user.

Related Views

- DBA_XML_VIEWS describes all XML views the database.
- USER_XML_VIEWS describes the XML views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML view
VIEW_NAME	VARCHAR2(30)		Name of the XML view
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2(256)		Name of the XML Schema element that is used for the view

See Also:

- ["DBA_XML_VIEWS"](#) on page 3-139
- ["USER_XML_VIEWS"](#) on page 3-187

AUDIT_ACTIONS

AUDIT_ACTIONS describes audit trail action type codes. This table can be used to map action type numbers to action type names.

Column	Datatype	NULL	Description
ACTION	NUMBER	NOT NULL	Numeric audit trail action type code. These values also appear in the COMMAND column of the V\$SESSION dynamic performance view, and are listed with that view. See Also: Table 5-5 on page 5-40 for a listing of the action type codes
NAME	VARCHAR2(28)	NOT NULL	Name of the type of audit trail action

CATALOG

CATALOG is included for compatibility. Oracle recommends that you not use this view.

CAT

CAT is a synonym for USER_CATALOG.

See Also: ["USER_CATALOG"](#) on page 3-161

CHAINED_ROWS

CHAINED_ROWS stores the output for the ANALYZE statement with the LIST CHAINED ROWS clause. You must run the utlchain.sql or utlchn1.sql script to create this table.

Column	Description
OWNER_NAME	Table owner
TABLE_NAME	Table name
CLUSTER_NAME	Cluster the table is in, if any
PARTITION_NAME	The name of the partition
SUBPARTITION_NAME	The name of the subpartition
HEAD_ROWID	ROWID the chained row is accessed by
ANALYZE_TIMESTAMP	Date/time that the ANALYZE statement was issued

CHANGE_SETS

CHANGE_SETS describes existing change sets. The user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers.

Column	Datatype	NULL	Description
SET_NAME	VARCHAR2(30)	NOT NULL	User-specified identifier. For synchronous change sets, there is one pre-created set with the SET_NAME "SYNC_SET".
CHANGE_SOURCE_NAME	VARCHAR2(30)	NOT NULL	Change source to which this set belongs
BEGIN_DATE	DATE		Starting point for capturing change data
END_DATE	DATE		Stopping point for capturing change data
BEGIN_SCN	NUMBER		Reserved for future use
END_SCN	NUMBER		Reserved for future use
FRESHNESS_DATE	DATE		Reserved for future use

CHANGE_SOURCES

Column	Datatype	NULL	Description
FRESHNESS_SCN	NUMBER		Reserved for future use
ADVANCE_ENABLED	VARCHAR2(1)		Indicates whether eligibility for advance (Y) or not (N)
IGNORE_DDL	VARCHAR2(1)		Obsolete column
CREATED	DATE	NOT NULL	Creation date of the change set
ROLLBACK_SEGMENT_NAME	VARCHAR2(30)		Reserved for future use
ADVANCING	VARCHAR2(1)	NOT NULL	Reserved for future use
PURGING	VARCHAR2(1)	NOT NULL	Indicates whether a purge of old data is in progress for this change set (set by advance) (Y) or not (N)
LOWEST_SCN	NUMBER		Current low-watermark for change data remaining in this set after the latest purge
TABLESPACE	VARCHAR2(30)		Reserved for future use
CAPTURE_ENABLED	VARCHAR2(1)	NOT NULL	Indicates whether capture is enabled (Y) or not (N)
STOP_ON_DDL	VARCHAR2(1)	NOT NULL	Indicates whether change set stops on DDL (Y) or not (N)
CAPTURE_ERROR	VARCHAR2(1)	NOT NULL	Indicates whether there is a capture error (Y) or not (N)
CAPTURE_NAME	VARCHAR2(30)		Streams capture process name
QUEUE_NAME	VARCHAR2(30)		Streams queue name
QUEUE_TABLE_NAME	VARCHAR2(30)		Streams queue table name
APPLY_NAME	VARCHAR2(30)		Streams apply process name
SET_DESCRIPTION	VARCHAR2(255)		Comment field for the publisher
PUBLISHER	VARCHAR2(30)		Publisher of the change set

CHANGE_SOURCES

CHANGE_SOURCES describes existing change sources. The user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers.

Column	Datatype	NULL	Description
SOURCE_NAME	VARCHAR2(30)	NOT NULL	User-specified identifier. For synchronous change sources, there is one pre-created source with the SOURCE_NAME "SYNC_SOURCE."
DBID	NUMBER		Reserved for future use
LOG_DIRECTORY	VARCHAR2(2000)		Reserved for future use
LOGFILE_PATTERN	VARCHAR2(30)		Reserved for future use
SOURCE_DESCRIPTION	VARCHAR2(255)		Comment field for customers
CREATED	DATE	NOT NULL	Creation date of the change source
SOURCE_TYPE	VARCHAR2(11)		Change source capture mode: <ul style="list-style-type: none">■ AUTOLOG■ HOTLOG■ SYNCHRONOUS
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database
FIRST_SCN	NUMBER		SCN of a LogMiner dictionary at which capture can begin
PUBLISHER	VARCHAR2(30)		Publisher of the change source

CHANGE_TABLES

CHANGE_TABLES describes existing change tables. The user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers.

Column	Datatype	NULL	Description
CHANGE_TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Change table owner
CHANGE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Change table name
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	The change set this table belongs to
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	The table owner in the source system
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	The name of the corresponding table in the source system
CREATED	DATE	NOT NULL	Date on which this change table was created
CREATED_SCN	NUMBER		System change number (SCN) when this table was created
CAPTURED_VALUES	VARCHAR2(1)	NOT NULL	Indicates the values O (old), N (new), or B (both)
PUB_ID	NUMBER	NOT NULL	Publication ID for subscribers

CLU

CLU is a synonym for USER_CLUSTERS.

See Also: ["USER_CLUSTERS"](#) on page 3-161

COL

COL is included for compatibility. Oracle recommends that you not use this view.

COLS

COLS is a synonym for USER_TAB_COLUMNS.

See Also: ["USER_TAB_COLUMNS"](#) on page 3-182

DATABASE_PROPERTIES

DATABASE_PROPERTIES lists Permanent database properties.

Column	Datatype	NULL	Description
PROPERTY_NAME	VARCHAR2(30)	NOT NULL	Property name
PROPERTY_VALUE	VARCHAR2(4000)		Property value
DESCRIPTION	VARCHAR2(4000)		Property description

Static Data Dictionary Views: DBA_2PC_NEIGHBORS to USER_XML_VIEWS

This chapter contains the static data dictionary views DBA_2PC_NEIGHBORS through USER_XML_VIEWS.

DBA_2PC_NEIGHBORS

DBA_2PC_NEIGHBORS describes incoming and outgoing connections for pending transactions.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)		Local identifier of a transaction
IN_OUT	VARCHAR2(3)		IN for incoming connections, OUT for outgoing
DATABASE	VARCHAR2(128)		IN for client database name, OUT for outgoing database link
DBUSER_OWNER	VARCHAR2(30)		IN for name of local user, OUT for owner of database link
INTERFACE	VARCHAR2(1)		C for request commit, otherwise N for prepare or request readonly commit
DBID	VARCHAR2(16)		Database ID at the other end of the connection
SESS#	NUMBER(38)		Session number of the connection at this database
BRANCH	VARCHAR2(128)		Transaction branch ID of the connection at this database

DBA_2PC_PENDING

DBA_2PC_PENDING describes distributed transactions awaiting recovery.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)	NOT NULL	String of form: n.n.n; n is a number
GLOBAL_TRAN_ID	VARCHAR2(169)		Globally unique transaction ID
STATE	VARCHAR2(16)	NOT NULL	Collecting, prepared, committed, forced commit, or forced rollback
MIXED	VARCHAR2(3)		YES indicates part of the transaction committed and part rolled back
ADVICE	VARCHAR2(1)		C for commit, R for rollback, else NULL
TRAN_COMMENT	VARCHAR2(255)		Text for commit work comment text
FAIL_TIME	DATE	NOT NULL	Value of SYSDATE when the row was inserted (transaction or system recovery)
FORCE_TIME	DATE		Time of manual force decision (null if not forced locally)
RETRY_TIME	DATE	NOT NULL	Time automatic recovery (RECO) last tried to recover the transaction
OS_USER	VARCHAR2(64)		Operating system-specific name for the end-user
OS_TERMINAL	VARCHAR2(255)		Operating system-specific name for the end-user terminal
HOST	VARCHAR2(128)		Name of the host machine for the end-user
DB_USER	VARCHAR2(30)		Oracle user name of the end-user at the topmost database
COMMIT#	VARCHAR2(16)		Global commit number for committed transactions

DBA_ADVISOR_ACTIONS

DBA_ADVISOR_ACTIONS displays information about the actions associated with all recommendations in the database. Each action is specified by the COMMAND and ATTR1 through ATTR6 columns. Each command defines how the attribute columns will be used.

Related View

USER_ADVISOR_ACTIONS displays information about the actions associated with the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
REC_ID	NUMBER	NOT NULL	Recommendation associated with the action
ACTION_ID	NUMBER	NOT NULL	Unique identifier for the action
OBJECT_ID	NUMBER		Object associated with the action
COMMAND	VARCHAR2(64)		Command to be executed See Also: DBA_ADVISOR_COMMANDS for a list of commands
COMMAND_ID	NUMBER	NOT NULL	ID of the command to be executed See Also: DBA_ADVISOR_COMMANDS for a list of commands
FLAGS	NUMBER		Advisor-specific flags
ATTR1	VARCHAR2(4000)		Parameters defining the command

Column	Datatype	NULL	Description
ATTR2	VARCHAR2(4000)		Parameters defining the command
ATTR3	VARCHAR2(4000)		Parameters defining the command
ATTR4	VARCHAR2(4000)		Parameters defining the command
ATTR5	CLOB		Parameters defining the command; to be used if the text is significantly large (for example, a SQL statement defining a materialized view)
ATTR6	CLOB		Parameters defining the command; to be used if the text is significantly large (for example, a SQL statement defining a materialized view)
NUM_ATTR1	NUMBER		General numeric attribute
NUM_ATTR2	NUMBER		General numeric attribute
NUM_ATTR3	NUMBER		General numeric attribute
NUM_ATTR4	NUMBER		General numeric attribute
NUM_ATTR5	NUMBER		General numeric attribute
MESSAGE	VARCHAR2(4000)		Message associated with the action

See Also: ["USER_ADVISOR_ACTIONS"](#) on page 3-157

DBA_ADVISOR_COMMANDS

DBA_ADVISOR_COMMANDS displays information about the commands used by all advisors in the database for specifying recommendation actions. In addition to the set of commands in the COMMAND column of V\$SESSION, the following additional commands are defined:

- RUN ADVISOR
- CHECK EXECUTION PLAN
- ALTER PARAMETER
- ENABLE TRACE

Column	Datatype	NULL	Description
COMMAND_ID	NUMBER		Identifier of the command
COMMAND_NAME	VARCHAR2(64)		Name of the command

DBA_ADVISOR_DEFINITIONS

DBA_ADVISOR_DEFINITIONS displays the properties of all advisors in the database. The view contains one row for each task, representing the current state of the task as well as execution-specific data such as progress monitoring and completion status.

Column	Datatype	NULL	Description
ADVISOR_ID	NUMBER	NOT NULL	Unique identifier for the advisor
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	Name of the advisor

Column	Datatype	NULL	Description
PROPERTY	NUMBER	NOT NULL	Properties: <ul style="list-style-type: none"> Bit 0: - Indicates whether the advisor runs in COMPREHENSIVE mode (1) or not (0) Bit 1: - Indicates whether the advisor runs in LIMITED mode (1) or not (0) Bit 2: - Indicates whether the advisor is resumable (1) or not (0) Bit 3: - Indicates whether the advisor accepts user directives (1) or not (0)

DBA_ADVISOR_FINDINGS

DBA_ADVISOR_FINDINGS displays the findings discovered by all advisors in the database.

Related View

USER_ADVISOR_FINDINGS displays the findings discovered by the advisors owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
FINDING_ID	NUMBER	NOT NULL	Identifier of the finding
TYPE	VARCHAR2(11)		Type of the finding: <ul style="list-style-type: none"> PROBLEM SYMPTOM ERROR INFORMATION
PARENT	NUMBER	NOT NULL	Identifier of the parent finding
OBJECT_ID	NUMBER		Identifier of the associated object, if any
IMPACT_TYPE	VARCHAR2(4000)		Impact of the finding on the system
IMPACT	NUMBER		Impact value
MESSAGE	VARCHAR2(4000)		Message describing the finding
MORE_INFO	VARCHAR2(4000)		Additional info associated with the finding

See Also: ["USER_ADVISOR_FINDINGS"](#) on page 3-157

DBA_ADVISOR_JOURNAL

DBA_ADVISOR_JOURNAL displays the journal entries for all tasks in the database.

Related View

USER_ADVISOR_JOURNAL displays the journal entries for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task

Column	Datatype	NULL	Description
TASK_ID	NUMBER	NOT NULL	Identifier of the task or workload object
TASK_NAME	VARCHAR2(30)		Name of the task or workload object
JOURNAL_ENTRY_SEQ	NUMBER	NOT NULL	Sequence number of the journal entry (unique for each task). This sequence number is used to order the data.
JOURNAL_ENTRY_TYPE	VARCHAR2(11)		Type of the task: <ul style="list-style-type: none"> ▪ INFORMATION ▪ WARNING ▪ ERROR ▪ FATAL
JOURNAL_ENTRY	VARCHAR2(4000)		Entry in the journal

See Also: "[USER_ADVISOR_JOURNAL](#)" on page 3-157

DBA_ADVISOR_LOG

DBA_ADVISOR_LOG displays information about the current state of all tasks in the database, as well as execution-specific data such as progress monitoring and completion status. The view contains one row for each task.

Related View

USER_ADVISOR_LOG displays information about the current state of the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_END	DATE		Execution end date and time of the task
STATUS	VARCHAR2(11)		Current operational status of the task: <ul style="list-style-type: none"> ▪ INITIAL - Initial state of the task; no recommendations are present ▪ EXECUTING - Task is currently running ▪ COMPLETED - Task successfully completed the analysis operation. Recommendation data can be viewed and reported. ▪ INTERRUPTED - Task analysis was interrupted by the user. Recommendation data, if present, can be viewed and reported at this time. ▪ CANCELLED ▪ FATAL ERROR - A fatal error occurred during the analysis operation. All recommendation data is unusable.
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor regarding the status
PCT_COMPLETION_TIME	NUMBER		Percent completion, in terms of time, of the task when it is executing
PROGRESS_METRIC	NUMBER		Metric that measures the progress of the task in terms of quality. Each advisor could have its own metric.
METRIC_UNITS	VARCHAR2(64)		Unit of the metric used to measure progress

Column	Datatype	NULL	Description
ACTIVITY_COUNTER	NUMBER		Counter that is updated frequently by the advisor, denoting that useful work is being performed
RECOMMENDATION_COUNT	NUMBER		Number of recommendations produced
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition

See Also: ["USER_ADVISOR_LOG"](#) on page 3-157

DBA_ADVISOR_OBJECT_TYPES

DBA_ADVISOR_OBJECT_TYPES displays information about the object types used by all advisors in the database. In addition to the regular database object types (such as TABLE and INDEX), the following types are defined:

- SYSTEM
- I/O
- SGA
- PGA
- SHARED POOL
- BUFFER CACHE
- LIBRARY CACHE
- PROCESS
- SESSION
- ENQUEUE
- LATCH
- ROLLBACK SEGMENT
- FILE
- PARAMETER
- CURSOR
- SQL
- SQL WORKLOAD

Column	Datatype	NULL	Description
OBJECT_TYPE_ID	NUMBER		Type identifier
OBJECT_TYPE	VARCHAR2(64)		Type name

DBA_ADVISOR_OBJECTS

DBA_ADVISOR_OBJECTS displays information about the objects currently referenced by all advisors in the database. Each row in the view pertains to an object instantiation.

Related View

USER_ADVISOR_OBJECTS displays information about the objects currently referenced by the advisors owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
OBJECT_ID	NUMBER	NOT NULL	Identifier of the object
TYPE	VARCHAR2(64)		Name of the type
TYPE_ID	NUMBER	NOT NULL	Type identifier number
TASK_ID	NUMBER	NOT NULL	Task referencing the object
TASK_NAME	VARCHAR2(30)		Name of the task
ATTR1	VARCHAR2(4000)		Attributes and identifier of the object
ATTR2	VARCHAR2(4000)		Attributes and identifier of the object
ATTR3	VARCHAR2(4000)		Attributes and identifier of the object
ATTR4	CLOB		Attributes and identifiers that cannot be expressed in the ATTR1, ATTR2, and ATTR3 columns
ATTR5	VARCHAR2(4000)		Attributes and identifier of the object

Note: The definition of the ATTR1, ATTR2, ATTR3, ATTR4, and ATTR5 columns depends on the advisors that are using the object. For example, the SQL object type defines the attribute columns as follows:

- ATTR1 contains the SQL hash value
- ATTR2 contains the SQL address (in the cursor cache)
- ATTR4 contains the SQL text

See Also: ["USER_ADVISOR_OBJECTS"](#) on page 3-157

DBA_ADVISOR_PARAMETERS

DBA_ADVISOR_PARAMETERS displays the parameters and their current values for all tasks in the database.

Related View

USER_ADVISOR_PARAMETERS displays the parameters and their current values for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task or workload object
TASK_ID	NUMBER	NOT NULL	Unique identifier number of the task or workload object
TASK_NAME	VARCHAR2(30)		Name of the task or workload object
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter

Column	Datatype	NULL	Description
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	Value of the parameter. Numeric parameter values are converted to a string equivalent. Possible keywords as values: <ul style="list-style-type: none"> ▪ ALL ▪ UNLIMITED ▪ UNUSED
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter: <ul style="list-style-type: none"> ▪ NUMBER - Numeric value ▪ STRING - String value. If the string contains special characters, it will be enclosed in single quotes. ▪ STRINGLIST - Comma-separated list of string elements. If a string element contains a comma or other special characters, then the element will be enclosed in single quotes. ▪ TABLE - Single table reference. A reference will contain a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported. ▪ TABLELIST - List of one or more comma-separated table references. A reference will contain schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported.

See Also: ["USER_ADVISOR_PARAMETERS"](#) on page 3-157

DBA_ADVISOR_RATIONALE

DBA_ADVISOR_RATIONALE displays information about the rationales for all recommendations in the database.

Related View

USER_ADVISOR_RATIONALE displays information about the rationales for the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
REC_ID	NUMBER		Recommendation associated with the rationale
RATIONALE_ID	NUMBER	NOT NULL	Unique identifier for the rationale
IMPACT_TYPE	VARCHAR2(4000)		Impact on the system due to the problem described in the rationale. The impact can be described in terms of time, cost, or % degradation.
IMPACT	NUMBER		Calculated impact value
MESSAGE	VARCHAR2(4000)		Message containing an overview of the rationale
OBJECT_ID	NUMBER		Identifier of an object specified in the DBA_ADVISOR_OBJECTS view

Column	Datatype	NULL	Description
TYPE	VARCHAR2(30)		Type of the rationale; defines what data exists in the attribute columns and how to interpret it: <ul style="list-style-type: none"> TEXT - Text sentence for descriptive messages. The ATTR1 column contains the text. CHART - Chart containing data to be displayed. The ATTR1 column contains the data.
ATTR1	VARCHAR2(4000)		Parameters defining the rationale
ATTR2	VARCHAR2(4000)		Parameters defining the rationale
ATTR3	VARCHAR2(4000)		Parameters defining the rationale
ATTR4	VARCHAR2(4000)		Parameters defining the rationale
ATTR5	CLOB		Parameters defining the rationale

See Also: ["USER_ADVISOR_RATIONALE"](#) on page 3-157

DBA_ADVISOR_RECOMMENDATIONS

DBA_ADVISOR_RECOMMENDATIONS displays the results of an analysis of all recommendations in the database. A recommendation can have multiple actions associated with it. Actions are described in the DBA_ADVISOR_ACTIONS view. A recommendation also points to a set of rationales that present a justification/reasoning for that recommendation. These rationales are in the DBA_ADVISOR_RATIONALE view.

Related View

USER_ADVISOR_RECOMMENDATIONS displays the results of an analysis of the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
REC_ID	NUMBER	NOT NULL	Unique identifier of the recommendation
TASK_ID	NUMBER	NOT NULL	Task that owns the recommendation
TASK_NAME	VARCHAR2(30)		Name of the task
FINDING_ID	NUMBER		Unique identifier of the finding
TYPE	VARCHAR2(30)		Type of the recommendation
RANK	NUMBER		Ranking, in terms of importance, within the set of recommendations generated for the task
PARENT_REC_IDS	VARCHAR2(4000)		Comma-separated list of the recommendation IDs of the parent recommendations. If this column is nonzero, then the recommendation depends on the parents, and cannot be accepted if the parents are not accepted.
BENEFIT_TYPE	VARCHAR2(4000)		Describes the benefit obtained by carrying out the recommendation If there is a set of parents for the recommendation, then the benefit is the cumulative benefit (the improvement in system performance when this and all prior parent recommendations are accepted). If there are no parents, then this is the improvement when the recommendation is accepted, independent of other recommendations.
BENEFIT	NUMBER		Calculated benefit value

Column	Datatype	NULL	Description
ANNOTATION_STATUS	VARCHAR2(11)		<p>When a task is complete, the recommendations are marked ACCEPT. The status can be changed later using the MARK_RECOMMENDATION procedure:</p> <ul style="list-style-type: none"> ▪ ACCEPT - Current recommendation is ready to implement. This recommendation can also be used as advice for future analysis operations. ▪ REJECT - Current recommendation is not acceptable to the user, and therefore will be excluded from any implementation scripts. This recommendation can also be used as advice for future analysis operations. ▪ IGNORE - Though not rejected, the current recommendation will be ignored when generating scripts and will never be used as advice to future analysis operations.

See Also: ["USER_ADVISOR_RECOMMENDATIONS"](#) on page 3-158

DBA_ADVISOR_SQLA_REC_SUM

DBA_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for all workload objects in the database after an Access Advisor analysis operation.

Related View

USER_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER		Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
REC_ID	NUMBER		Identifier of the recommendation
TOTAL_STMTS	NUMBER		Total number of statements processed during analysis
TOTAL_PRECOST	NUMBER		Total cost of executing the statements in which the recommended object will be utilized, prior to the recommendations
TOTAL_POSTCOST	NUMBER		Total cost of executing the statements in which the recommended object will be utilized, after the recommendations have been implemented

See Also: ["USER_ADVISOR_SQLA_REC_SUM"](#) on page 3-158

DBA_ADVISOR_SQLA_WK_MAP

DBA_ADVISOR_SQLA_WK_MAP displays the workload references for all tasks in the database. Workload references are necessary to allow the SQL Access Advisor to find required workload data.

Related View

USER_ADVISOR_SQLA_WK_MAP displays the workload references for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER		Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
WORKLOAD_ID	NUMBER		Unique identifier of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload

See Also: ["USER_ADVISOR_SQLA_WK_MAP"](#) on page 3-158

DBA_ADVISOR_SQLA_WK_STMTS

DBA_ADVISOR_SQLA_WK_STMTS displays information about all workload objects in the database after an Access Advisor analysis operation.

Related View

USER_ADVISOR_SQLA_WK_STMTS displays information about the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
SQL_ID	NUMBER	NOT NULL	Generated identifier of the statement
HASH_VALUE	NUMBER		Hash value for the parent statement in the SQL cache
USERNAME	VARCHAR2(30)		Name of the user executing the statement
MODULE	VARCHAR2(64)		Name of the module issuing the statement
ACTION	VARCHAR2(64)		Module action for the statement
CPU_TIME	NUMBER		Total CPU count (in seconds) of the executing statement
BUFFER_GETS	NUMBER		Total number of buffer gets for the statement
DISK_READS	NUMBER		Total disk-read I/O count for the statement
ELAPSED_TIME	NUMBER		Total elapsed time (in seconds) of the executing statement
ROWS_PROCESSED	NUMBER		Total number of rows processed by the statement
EXECUTIONS	NUMBER		Total number of times the statement was executed
PREL_COST	NUMBER		Cost of executing the statement in the workload prior to the recommendations
POSTL_COST	NUMBER		Cost of executing the statement in the workload after the recommendations
LAST_EXECUTION_DATE	DATE		Date on which the statement was last executed
PRIORITY	NUMBER		Business importance of the statement: <ul style="list-style-type: none"> ■ 1 - High ■ 2 - Medium ■ 3 - Low
COMMAND_TYPE	NUMBER		Type of the command
STAT_PERIOD	NUMBER		Unused

Column	Datatype	NULL	Description
SQL_TEXT	CLOB		Text of the SQL statement
IMPORTANCE	NUMBER		Advisor-calculated importance value
REC_ID	NUMBER		Associated recommendation identifier

See Also: ["USER_ADVISOR_SQLA_WK_STMTS"](#) on page 3-158

DBA_ADVISOR_SQLW_JOURNAL

DBA_ADVISOR_SQLW_JOURNAL displays the journal entries for all workload objects in the database.

Related View

USER_ADVISOR_SQLW_JOURNAL displays the journal entries for the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload
WORKLOAD_ID	NUMBER	NOT NULL	Identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload object
JOURNAL_ENTRY_SEQ	NUMBER	NOT NULL	Sequence number of the journal entry (unique for each workload). The sequence number is used to order the data.
JOURNAL_ENTRY_TYPE	VARCHAR2(13)		Type of the task: <ul style="list-style-type: none"> ▪ INFORMATIONAL ▪ WARNING ▪ ERROR ▪ FATAL
JOURNAL_ENTRY	VARCHAR2(4000)		Entry in the journal

See Also: ["USER_ADVISOR_SQLW_JOURNAL"](#) on page 3-158

DBA_ADVISOR_SQLW_PARAMETERS

DBA_ADVISOR_SQLW_PARAMETERS displays all workload parameters and their current values in the database.

Related View

USER_ADVISOR_SQLW_PARAMETERS displays the workload parameters and their current values owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task or workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload object
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter

Column	Datatype	NULL	Description
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	Value of the parameter. Numeric parameter values are converted to a string equivalent. Possible keywords as values: <ul style="list-style-type: none"> ▪ ALL ▪ UNLIMITED ▪ UNUSED
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter: <ul style="list-style-type: none"> ▪ NUMBER - Numeric value ▪ STRING - String value. If the string contains special characters, then it will be enclosed in single quotes. ▪ STRINGLIST - Comma-separated list of string elements. If a string element contains a comma or other special characters, then the element will be enclosed in single quotes. ▪ TABLE - Single table reference. A reference contains a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported. ▪ TABLELIST - List of one or more comma-separated table references. A reference contains a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported.

See Also: ["USER_ADVISOR_SQLW_PARAMETERS"](#) on page 3-158

DBA_ADVISOR_SQLW_STMTS

DBA_ADVISOR_SQLW_STMTS displays rows that correspond to all statements in the workload. All columns are guaranteed to be non-null.

Related View

USER_ADVISOR_SQLW_STMTS displays rows that correspond to the statements in the workload owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
SQL_ID	NUMBER	NOT NULL	Generated identifier of the statement
HASH_VALUE	NUMBER		Hash value for the parent statement in the cache
USERNAME	VARCHAR2(30)		Name of the user executing the statement
MODULE	VARCHAR2(64)		Name of the module issuing the statement
ACTION	VARCHAR2(64)		Module action for the statement
CPU_TIME	NUMBER		Total CPU count (in seconds) of the executing statement
BUFFER_GETS	NUMBER		Total number of buffer gets for the statement
DISK_READS	NUMBER		Total disk-read I/O count for the statement

Column	Datatype	NULL	Description
ELAPSED_TIME	NUMBER		Total elapsed time (in seconds) of the executing statement
ROWS_PROCESSED	NUMBER		Total number of rows processed by the statement
EXECUTIONS	NUMBER		Total number of times the statement was executed
OPTIMIZER_COST	NUMBER		Cost of executing the statement in the workload prior to the recommendations
LAST_EXECUTION_DATE	DATE		Date on which the statement was last executed
PRIORITY	NUMBER		Priority of the statement: <ul style="list-style-type: none"> ▪ 1 - High ▪ 2 - Medium ▪ 3 - Low
COMMAND_TYPE	NUMBER		Type of the command
STAT_PERIOD	NUMBER		Unused
SQL_TEXT	CLOB		Text of the SQL statement
VALID	NUMBER		Indicates whether the statement is valid for analysis: <ul style="list-style-type: none"> ▪ 0 - Statement will not be analyzed by the EXECUTE_TASK procedure. Typically, the statement references one or more tables that do not have valid statistics. To correct this problem, ensure that the tables have valid statistics and execute the RESET_SQLWKLD procedure on the current workload. ▪ 1 - Statement is eligible for analysis by the EXECUTE_TASK procedure.

See Also: ["USER_ADVISOR_SQLW_STMTS"](#) on page 3-158

DBA_ADVISOR_SQLW_SUM

DBA_ADVISOR_SQLW_SUM displays an aggregated picture of all SQLWkld workload objects in the database.

Related View

USER_ADVISOR_SQLW_SUM displays an aggregated picture of the SQLWkld workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Unique name of the workload
DESCRIPTION	VARCHAR2(256)		User-specified description of the workload
CREATE_DATE	DATE	NOT NULL	Date on which the workload object was created
MODIFY_DATE	DATE	NOT NULL	Date of last update to the current workload
NUM_SELECT_STMT	NUMBER		Number of SELECT statements in the workload
NUM_UPDATE_STMT	NUMBER		Number of UPDATE statements in the workload
NUM_DELETE_STMT	NUMBER		Number of DELETE statements in the workload
NUM_INSERT_STMT	NUMBER		Number of INSERT statements in the workload
NUM_MERGE_STMT	NUMBER		Number of MERGE statements in the workload
SOURCE	VARCHAR2(30)		Optional name that identifies the creator of the object

Column	Datatype	NULL	Description
HOW_CREATED	VARCHAR2 (30)		Optional object or template on which the object was based
DATA_SOURCE	VARCHAR2 (2000)		Workload data source
READ_ONLY	VARCHAR2 (5)		Indicates whether or not the workload can be modified or deleted (TRUE) or not (FALSE)

See Also: ["USER_ADVISOR_SQLW_SUM"](#) on page 3-159

DBA_ADVISOR_SQLW_TABLES

DBA_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement.

Related View

USER_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the workload object
WORKLOAD_ID	NUMBER		Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2 (30)		Name of the workload
SQL_ID	NUMBER		Identifier of the statement
TABLE_OWNER	VARCHAR2 (30)		Owner of the table
TABLE_NAME	VARCHAR2 (30)		Name of the table

See Also: ["USER_ADVISOR_SQLW_TABLES"](#) on page 3-159

DBA_ADVISOR_SQLW_TEMPLATES

DBA_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of all SQLWkld template objects in the database.

Related View

USER_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of the SQLWkld template objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2 (30)		Unique name of the workload
DESCRIPTION	VARCHAR2 (256)		User-specified description of the workload
CREATE_DATE	DATE	NOT NULL	Date on which the workload object was created
MODIFY_DATE	DATE	NOT NULL	Date of last update to the current workload
SOURCE	VARCHAR2 (30)		Optional object or template on which the object was based

Column	Datatype	NULL	Description
READ_ONLY	VARCHAR2(5)		Indicates whether the workload template can be modified or deleted (TRUE) or not (FALSE)

See Also: ["USER_ADVISOR_SQLW_TEMPLATES"](#) on page 3-159

DBA_ADVISOR_TASKS

DBA_ADVISOR_TASKS displays information about all tasks in the database. The view contains one row for each task. Each task contains a name that is unique to the owner. Task names are just informational and no uniqueness is enforced within any namespace.

Related View

USER_ADVISOR_TASKS displays information about the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
CREATED	DATE	NOT NULL	Creation date of the task
LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
PARENT_TASK_ID	NUMBER		Identifier of the parent task (if the task was created as a result of the recommendation of another task)
PARENT_REC_ID	NUMBER		Identifier of the recommendation within the parent task that resulted in the creation of the task
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_END	DATE		Execution end date and time of the task
STATUS	VARCHAR2(11)		Current operational status of the task: <ul style="list-style-type: none"> ■ INITIAL - Initial state of the task; no recommendations are present ■ EXECUTING - Task is currently running ■ INTERRUPTED - Task analysis was interrupted by the user. Recommendation data, if present, can be viewed and reported at this time. ■ COMPLETED - Task successfully completed the analysis operation. Recommendation data can be viewed and reported. ■ ERROR - An error occurred during the analysis operation. Recommendations, if present, can be viewed and reported at this time.
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor regarding the status
PCT_COMPLETION_TIME	NUMBER		Percent completion, in terms of time, of the task when it is executing
PROGRESS_METRIC	NUMBER		Metric that measures the progress of the task in terms of quality. Each advisor may have its own metric.
METRIC_UNITS	VARCHAR2(64)		Unit of the metric used to measure progress

Column	Datatype	NULL	Description
ACTIVITY_COUNTER	NUMBER		Counter that is updated frequently by the advisor, denoting that useful work is being performed
RECOMMENDATION_COUNT	NUMBER		Number of recommendations produced
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition
SOURCE	VARCHAR2(30)		Optional name that identifies the creator of the task
HOW_CREATED	VARCHAR2(30)		Optional task or template on which the object was based
READ_ONLY	VARCHAR2(5)		Indicates whether the task is read-only (TRUE) or not (FALSE)

See Also: ["USER_ADVISOR_TASKS"](#) on page 3-159

DBA_ADVISOR_TEMPLATES

DBA_ADVISOR_TEMPLATES displays information about all templates in the database.

Related View

USER_ADVISOR_TEMPLATES displays information about the templates owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
CREATED	DATE	NOT NULL	Creation date of the task
LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
SOURCE	VARCHAR2(30)		Optional task or template on which the template was based
READ_ONLY	VARCHAR2(5)		Indicates whether the task can be modified or deleted (TRUE) or not (FALSE)

See Also: ["USER_ADVISOR_TEMPLATES"](#) on page 3-159

DBA_ADVISOR_USAGE

DBA_ADVISOR_USAGE displays the usage information for each type of advisor in the database.

Column	Datatype	NULL	Description
ADVISOR_ID	NUMBER	NOT NULL	Type of the advisor
LAST_EXEC_TIME	DATE	NOT NULL	Date of the last execution
NUM_EXECS	NUMBER	NOT NULL	Cumulative number of executions

DBA_ALERT_HISTORY

DBA_ALERT_HISTORY describes a time-limited history of alerts which are no longer outstanding.

Column	Datatype	NULL	Description
SEQUENCE_ID	NUMBER	NOT NULL	Alert sequence number
REASON_ID	NUMBER		ID of the alert reason
OWNER	VARCHAR2(30)		Owner of the object on which the alert was issued
OBJECT_NAME	VARCHAR2(513)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(64)		Object type
REASON	VARCHAR2(4000)		Reason for the alert
TIME_SUGGESTED	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was last updated
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was first produced
SUGGESTED_ACTION	VARCHAR2(4000)		Advice of the recommended action
ADVISOR_NAME	VARCHAR2(30)		Name of the advisor to be invoked for more information
METRIC_VALUE	NUMBER		Value of the related metrics
MESSAGE_TYPE	VARCHAR2(12)		Message type: <ul style="list-style-type: none"> ▪ Notification ▪ Warning
MESSAGE_GROUP	VARCHAR2(64)		Name of the message group to which the alert belongs
MESSAGE_LEVEL	NUMBER		Severity message level (1 to 32)
HOSTING_CLIENT_ID	VARCHAR2(64)		ID of the client or security group to which the alert relates
MODULE_ID	VARCHAR2(64)		ID of the module that originated the alert
PROCESS_ID	VARCHAR2(128)		Process ID
HOST_ID	VARCHAR2(256)		DNS host name of the originating host
HOST_NW_ADDR	VARCHAR2(256)		IP or other network address of the originating host
INSTANCE_NAME	VARCHAR2(15)		Originating instance name
INSTANCE_NUMBER	NUMBER		Originating instance number
USER_ID	VARCHAR2(30)		User ID
EXECUTION_CONTEXT_ID	VARCHAR2(60)		ID of the thread of execution
ERROR_INSTANCE_ID	VARCHAR2(142)		ID of an error instance plus a sequence number
RESOLUTION	VARCHAR2(7)		Resolution: <ul style="list-style-type: none"> ▪ cleared ▪ N/A

DBA_ALL_TABLES

DBA_ALL_TABLES describes all object tables and relational tables in the database. Its columns are the same as those in ALL_ALL_TABLES.

See Also: ["ALL_ALL_TABLES"](#) on page 2-4

DBA_APPLICATION_ROLES

DBA_APPLICATION_ROLES describes all the roles that have authentication policy functions defined.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the application role
SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the authorized package
PACKAGE	VARCHAR2(30)	NOT NULL	Name of the authorized package

DBA_APPLY

DBA_APPLY displays information about all apply processes in the database. Its columns are the same as those in ALL_APPLY.

See Also: ["ALL_APPLY"](#) on page 2-7

DBA_APPLY_CONFLICT_COLUMNS

DBA_APPLY_CONFLICT_COLUMNS displays information about conflict handlers on all tables in the database. Its columns are the same as those in ALL_APPLY_CONFLICT_COLUMNS.

See Also: ["ALL_APPLY_CONFLICT_COLUMNS"](#) on page 2-8

DBA_APPLY_DML_HANDLERS

DBA_APPLY_DML_HANDLERS displays information about the DML handlers on all tables in the database. Its columns are the same as those in ALL_APPLY_DML_HANDLERS.

See Also: ["ALL_APPLY_DML_HANDLERS"](#) on page 2-8

DBA_APPLY_ENQUEUE

DBA_APPLY_ENQUEUE displays information about the apply enqueue actions for all rules in the database. Its columns are the same as those in ALL_APPLY_ENQUEUE.

See Also: ["ALL_APPLY_ENQUEUE"](#) on page 2-9

DBA_APPLY_ERROR

DBA_APPLY_ERROR displays information about error transactions generated by all apply processes in the database. Its columns are the same as those in ALL_APPLY_ERROR.

See Also: ["ALL_APPLY_ERROR"](#) on page 2-9

DBA_APPLY_EXECUTE

DBA_APPLY_EXECUTE displays information about the apply execute actions for all rules in the database. Its columns are the same as those in ALL_APPLY_EXECUTE.

See Also: ["ALL_APPLY_EXECUTE"](#) on page 2-10

DBA_APPLY_INSTANTIATED_GLOBAL

DBA_APPLY_INSTANTIATED_GLOBAL displays information about databases for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database that was instantiated
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the database. Only changes committed after this SCN are applied by an apply process.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_INSTANTIATED_OBJECTS

DBA_APPLY_INSTANTIATED_OBJECTS displays information about objects for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database where the object originated
SOURCE_OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object at the source database
SOURCE_OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object at the source database
SOURCE_OBJECT_TYPE	VARCHAR2(11)		Type of the object at the source database
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the object. Only changes committed after this SCN are applied by an apply process.
IGNORE_SCN	NUMBER		SCN below which the instantiation SCN cannot be set. This value corresponds to the SCN value at the source database at the time when the object was prepared for instantiation.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_INSTANTIATED_SCHEMAS

DBA_APPLY_INSTANTIATED_SCHEMAS displays information about schemas for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database where the schema originated
SOURCE_SCHEMA	VARCHAR2(30)		Name of the schema at the source database
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the schema. Only changes committed after this SCN are applied by an apply process.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_KEY_COLUMNS

DBA_APPLY_KEY_COLUMNS displays information about substitute key columns for all tables in the database. Its columns are the same as those in ALL_APPLY_KEY_COLUMNS.

See Also: ["ALL_APPLY_KEY_COLUMNS"](#) on page 2-10

DBA_APPLY_PARAMETERS

DBA_APPLY_PARAMETERS displays information about the parameters for all apply processes in the database. Its columns are the same as those in ALL_APPLY_PARAMETERS.

See Also: ["ALL_APPLY_PARAMETERS"](#) on page 2-10

DBA_APPLY_PROGRESS

DBA_APPLY_PROGRESS displays information about the progress made by all apply processes in the database. Its columns are the same as those in ALL_APPLY_PROGRESS.

See Also: ["ALL_APPLY_PROGRESS"](#) on page 2-11

DBA_APPLY_TABLE_COLUMNS

DBA_APPLY_TABLE_COLUMNS displays information about the destination table object columns for all tables in the database. Its columns are the same as those in ALL_APPLY_TABLE_COLUMNS.

See Also: ["ALL_APPLY_TABLE_COLUMNS"](#) on page 2-11

DBA_AQ_AGENT_PRIVS

DBA_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to all users in the database.

Related View

USER_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to the current user. This view does not display the DB_USERNAME column.

Column	Datatype	NULL	Description
AGENT_NAME	VARCHAR2(30)	NOT NULL	Name of the AQ agent
DB_USERNAME	VARCHAR2(30)		Name of the database user that the agent maps to
HTTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through HTTP (YES) or not (NO)
SMTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through SMTP (YES) or not (NO)

See Also: ["USER_AQ_AGENT_PRIVS"](#) on page 3-159

DBA_AQ_AGENTS

DBA_AQ_AGENTS displays information about all registered AQ agents in the database.

Column	Datatype	NULL	Description
AGENT_NAME	VARCHAR2(30)	NOT NULL	Name of the AQ agent
HTTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through HTTP (YES) or not (NO)
SMTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through SMTP (YES) or not (NO)

DBA_ASSOCIATIONS

DBA_ASSOCIATIONS describes all user-defined statistics in the database. Its columns are the same as those in "[ALL_ASSOCIATIONS](#)" on page 2-13.

DBA_ATTRIBUTE_TRANSFORMATIONS

DBA_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for all transformations in the database.

Related View

USER_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for the transformations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TRANSFORMATION_ID	NUMBER	NOT NULL	Unique identifier for the transformation
OWNER	VARCHAR2(30)	NOT NULL	Owning user of the transformation
NAME	VARCHAR2(30)	NOT NULL	Transformation name
FROM_TYPE	VARCHAR2(61)		Source type name
TO_TYPE	VARCHAR2(91)		Target type name
ATTRIBUTE	NUMBER	NOT NULL	Target type attribute number
ATTRIBUTE_TRANSFORMATION	VARCHAR2(4000)		Transformation function for the attribute

See Also: "[USER_ATTRIBUTE_TRANSFORMATIONS](#)" on page 3-160

DBA_AUDIT_EXISTS

DBA_AUDIT_EXISTS displays audit trail entries produced by AUDIT EXISTS and AUDIT NOT EXISTS.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Intended creator of the non-existent object
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object

Column	Datatype	NULL	Description
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: <ul style="list-style-type: none"> ■ 0 - Action succeeded ■ 2004 - Security violation
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: ["AUDIT_TRAIL"](#) on page 1-21

DBA_AUDIT_OBJECT

DBA_AUDIT_OBJECT displays audit trail records for all objects in the database.

Related View

USER_AUDIT_OBJECT displays audit trail records for all objects accessible to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both).
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: <ul style="list-style-type: none"> ■ 0 - Action succeeded ■ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query

Note: The `SQL_BIND` and `SQL_TEXT` columns are only populated if the `AUDIT_TRAIL` initialization parameter is set to `db_extended`.

See Also: ["AUDIT_TRAIL"](#) on page 1-21

DBA_AUDIT_POLICIES

`DBA_AUDIT_POLICIES` describes all fine-grained auditing policies in the database. Its columns are the same as those in `ALL_AUDIT_POLICIES`.

See Also: ["ALL_AUDIT_POLICIES"](#) on page 2-14

DBA_AUDIT_POLICY_COLUMNS

`DBA_AUDIT_POLICY_COLUMNS` describes all fine-grained auditing policy columns in the database. Its columns are the same as those in `ALL_AUDIT_POLICY_COLUMNS`.

See Also: ["ALL_AUDIT_POLICY_COLUMNS"](#) on page 2-14

DBA_AUDIT_SESSION

`DBA_AUDIT_SESSION` displays all audit trail records concerning `CONNECT` and `DISCONNECT`.

Related View

`USER_AUDIT_SESSION` contains audit trail records concerning connections and disconnections of the current user.

Column	Datatype	NULL	Description
<code>OS_USERNAME</code>	<code>VARCHAR2(255)</code>		Operating system login username of the user whose actions were audited
<code>USERNAME</code>	<code>VARCHAR2(30)</code>		Name (not ID number) of the user whose actions were audited
<code>USERHOST</code>	<code>VARCHAR2(128)</code>		Client host machine name
<code>TERMINAL</code>	<code>VARCHAR2(255)</code>		Identifier of the user's terminal
<code>TIMESTAMP</code>	<code>DATE</code>		Date and time of the creation of the audit trail entry (date and time of user login for entries created by <code>AUDIT_SESSION</code>) in the local database session time zone
<code>ACTION_NAME</code>	<code>VARCHAR2(28)</code>		Name of the action type corresponding to the numeric code in the <code>ACTION</code> column in <code>DBA_AUDIT_TRAIL</code>
<code>LOGOFF_TIME</code>	<code>DATE</code>		Date and time of user log off
<code>LOGOFF_LREAD</code>	<code>NUMBER</code>		Logical reads for the session
<code>LOGOFF_PREAD</code>	<code>NUMBER</code>		Physical reads for the session
<code>LOGOFF_LWRITE</code>	<code>NUMBER</code>		Logical writes for the session
<code>LOGOFF_DLOCK</code>	<code>VARCHAR2(40)</code>		Deadlocks detected during the session
<code>SESSIONID</code>	<code>NUMBER</code>	NOT NULL	Numeric ID for each Oracle session
<code>RETURNCODE</code>	<code>NUMBER</code>	NOT NULL	Oracle error code generated by the action. Some useful values: <ul style="list-style-type: none"> ▪ 0 - Action succeeded ▪ 2004 - Security violation

Column	Datatype	NULL	Description
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process

DBA_AUDIT_STATEMENT

DBA_AUDIT_STATEMENT displays audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements throughout the database.

Related View

USER_AUDIT_STATEMENT displays audit trail records for the same statements issued by the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with the ADMIN option
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement

Column	Datatype	NULL	Description
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both).
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail, inserted by the application
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: <ul style="list-style-type: none"> ■ 0 - Action succeeded ■ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: ["AUDIT_TRAIL"](#) on page 1-21

DBA_AUDIT_TRAIL

DBA_AUDIT_TRAIL displays all audit trail entries.

Related View

USER_AUDIT_TRAIL displays all audit trail entries related to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by <code>AUDIT SESSION</code>) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION	NUMBER	NOT NULL	Numeric action type code. The corresponding name of the action type is in the <code>ACTION_NAME</code> column.
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the <code>ACTION</code> column
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the <code>NEW_NAME</code> column
NEW_NAME	VARCHAR2(128)		New name of the object after a <code>RENAME</code> or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a <code>GRANT</code> or <code>REVOKE</code> statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a <code>GRANT</code> or <code>REVOKE</code> statement
ADMIN_OPTION	VARCHAR2(1)		Indicates whether the role or system privilege was granted with the <code>ADMIN</code> option
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a <code>GRANT</code> or <code>REVOKE</code> statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the <code>AUDIT</code> statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order <code>ALTER</code> , <code>AUDIT</code> , <code>COMMENT</code> , <code>DELETE</code> , <code>GRANT</code> , <code>INDEX</code> , <code>INSERT</code> , <code>LOCK</code> , <code>RENAME</code> , <code>SELECT</code> , <code>UPDATE</code> , <code>REFERENCES</code> , and <code>EXECUTE</code>). Positions 14, 15, and 16 are reserved for future use. The characters are: <ul style="list-style-type: none"> ■ - - None ■ S - Success ■ F - Failure ■ B - Both
LOGOFF_TIME	DATE		Date and time of user log off
LOGOFF_LREAD	NUMBER		Logical reads for the session
LOGOFF_PREAD	NUMBER		Physical reads for the session
LOGOFF_LWRITE	NUMBER		Logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Deadlocks detected during the session
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail entry, providing more information about the statement audited Also indicates how the user was authenticated. The method can be one of the following: <ul style="list-style-type: none"> ■ DATABASE - Authentication was done by password ■ NETWORK - Authentication was done by Oracle Net Services or the Advanced Security option ■ PROXY - Client was authenticated by another user; the name of the proxy user follows the method type

Column	Datatype	NULL	Description
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: <ul style="list-style-type: none"> ■ 0 - Action succeeded ■ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: ["AUDIT_TRAIL"](#) on page 1-21

DBA_AW_PS

DBA_AW_PS describes the pagespaces in all analytic workspaces in the database. Its columns are the same as those in ALL_AW_PS.

See Also: ["ALL_AW_PS"](#) on page 2-15

DBA_AWS

DBA_AWS describes all analytic workspaces in the database. Its columns are the same as those in ALL_AWS.

See Also: ["ALL_AWS"](#) on page 2-15

DBA_BASE_TABLE_MVIEWS

DBA_BASE_TABLE_MVIEWS describes all materialized views using materialized view logs in the database. Its columns are the same as those in ALL_BASE_TABLE_MVIEWS.

See Also: ["ALL_BASE_TABLE_MVIEWS"](#) on page 2-16

DBA_BLOCKERS

DBA_BLOCKERS displays a session if it is not waiting for a locked object but is holding a lock on an object for which another session is waiting.

Column	Datatype	NULL	Description
HOLDING_SESSION	NUMBER		Session holding a lock

DBA_CAPTURE

DBA_CAPTURE displays information about all capture processes in the database. Its columns are the same as those in ALL_CAPTURE.

See Also: ["ALL_CAPTURE"](#) on page 2-16

DBA_CAPTURE_EXTRA_ATTRIBUTES

DBA_CAPTURE_EXTRA_ATTRIBUTES displays information about the extra attributes for all capture processes in the database. Its columns are the same as those in ALL_CAPTURE_EXTRA_ATTRIBUTES.

See Also: ["ALL_CAPTURE_EXTRA_ATTRIBUTES"](#) on page 2-18

DBA_CAPTURE_PARAMETERS

DBA_CAPTURE_PARAMETERS displays information about the parameters for all capture processes in the database. Its columns are the same as those in ALL_CAPTURE_PARAMETERS.

See Also: ["ALL_CAPTURE_PARAMETERS"](#) on page 2-18

DBA_CAPTURE_PREPARED_DATABASE

DBA_CAPTURE_PREPARED_DATABASE displays information about when the local database was prepared for instantiation. Its columns are the same as those in ALL_CAPTURE_PREPARED_DATABASE.

See Also: ["ALL_CAPTURE_PREPARED_DATABASE"](#) on page 2-19

DBA_CAPTURE_PREPARED_SCHEMAS

DBA_CAPTURE_PREPARED_SCHEMAS displays information about all schemas prepared for instantiation at the local database. Its columns are the same as those in ALL_CAPTURE_PREPARED_SCHEMAS.

See Also: ["ALL_CAPTURE_PREPARED_SCHEMAS"](#) on page 2-19

DBA_CAPTURE_PREPARED_TABLES

DBA_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation at the local database. Its columns are the same as those in ALL_CAPTURE_PREPARED_TABLES.

See Also: ["ALL_CAPTURE_PREPARED_TABLES"](#) on page 2-19

DBA_CATALOG

DBA_CATALOG lists all indexes, tables, views, clusters, synonyms, and sequences in the database. Its columns are the same as those in ["ALL_CATALOG"](#) on page 2-20.

DBA_CLU_COLUMNS

DBA_CLU_COLUMNS maps all table columns to related cluster columns.

Related View

USER_CLU_COLUMNS maps all table columns owned by the current user to related cluster columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Cluster name
CLU_COLUMN_NAME	VARCHAR2(30)	NOT NULL	Key column in the cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Clustered table name
TAB_COLUMN_NAME	VARCHAR2(4000)		Key column or attribute of the object type column

DBA_CLUSTER_HASH_EXPRESSIONS

DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database. Its columns are the same as those in ["ALL_CLUSTER_HASH_EXPRESSIONS"](#) on page 2-20.

DBA_CLUSTERS

DBA_CLUSTERS describes all clusters in the database. Its columns are the same as those in ["ALL_CLUSTERS"](#) on page 2-20.

DBA_COL_COMMENTS

DBA_COL_COMMENTS displays comments on the columns of all tables and views in the database. Its columns are the same as those in ALL_COL_COMMENTS.

See Also: ["ALL_COL_COMMENTS"](#) on page 2-21

DBA_COL_PRIVS

DBA_COL_PRIVS describes all column object grants in the database.

Related View

USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: ["USER_COL_PRIVS"](#) on page 3-162

DBA_COLL_TYPES

DBA_COLL_TYPES describes all named collection types (arrays, nested tables, object tables, and so on) in the database. Its columns are the same as those in ["ALL_COLL_TYPES"](#) on page 2-23.

DBA_COMMON_AUDIT_TRAIL

DBA_COMMON_AUDIT_TRAIL displays all standard and fine-grained audit trail entries.

Column	Datatype	NULL	Description
AUDIT_TYPE	VARCHAR2(18)		Audit trail type: <ul style="list-style-type: none"> ■ Standard Audit ■ Fine Grained Audit
SESSION_ID	NUMBER		Numeric ID for the Oracle session
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
STATEMENTID	NUMBER		Numeric ID for the statement run; a statement may cause multiple audit records
ENTRYID	NUMBER		Numeric ID for the audit trail entry in the session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the audited operation (timestamp of user login for entries created by AUDIT SESSION) in the session's time zone
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
DB_USER	VARCHAR2(30)		Database user name of the user whose actions were audited
CLIENT_ID	VARCHAR2(64)		Client identifier in the Oracle session
EXT_NAME	VARCHAR2(4000)		User external name
OS_USER	VARCHAR2(255)		Operating system login user name of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
OS_PROCESS	VARCHAR2(16)		Operating system process identifier of the Oracle process

Column	Datatype	NULL	Description
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OBJECT_SCHEMA	VARCHAR2(30)		Owner of the audited object
OBJECT_NAME	VARCHAR2(128)		Name of the object affected by the action
POLICY_NAME	VARCHAR2(30)		Name of the Fine-Grained Auditing Policy
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of the object after a RENAME or the name of the underlying object
ACTION	NUMBER		Numeric action type code. The corresponding name of the action type is in the STATEMENT_TYPE column.
STATEMENT_TYPE	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object was accessed or modified
RETURNCODE	NUMBER		Oracle error code generated by the action (0 if the action succeeded)
SCN	NUMBER		System change number (SCN) of the query
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail entry, providing more information about the statement audited Also indicates how the user was authenticated: <ul style="list-style-type: none"> ■ DATABASE - Authentication was done by password ■ NETWORK - Authentication was done by Oracle Net Services or the Advanced Networking Option ■ PROXY - Client was authenticated by another user. The name of the proxy user follows the method type.
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Indicates whether the role or system privilege was granted with the ADMIN option
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: <ul style="list-style-type: none"> ■ - - None ■ S - Success ■ F - Failure ■ B - Both
LOGOFF_TIME	DATE		Timestamp of user log off
LOGOFF_LREAD	NUMBER		Number of logical reads in the session
LOGOFF_PREAD	NUMBER		Number of physical reads in the session

Column	Datatype	NULL	Description
LOGOFF_LWRITE	NUMBER		Number of logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Number of deadlocks detected during the session
SESSION_CPU	NUMBER		Amount of CPU time used by the Oracle session

DBA_CONS_OBJ_COLUMNS

DBA_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in all tables in the database. Its columns are the same as those in ALL_CONS_OBJ_COLUMNS.

See Also: ["ALL_CONS_OBJ_COLUMNS"](#) on page 2-24

DBA_CONSTRAINTS

DBA_CONSTRAINTS describes all constraint definitions on all tables in the database. Its columns are the same as those in ["ALL_CONSTRAINTS"](#) on page 2-25.

DBA_CONTEXT

DBA_CONTEXT provides all context namespace information in the database. Its columns are the same as those in ["ALL_CONTEXT"](#) on page 2-26.

DBA_DATA_FILES

DBA_DATA_FILES describes database files.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database file
FILE_ID	NUMBER	NOT NULL	File identifier number of the database file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file in bytes
BLOCKS	NUMBER	NOT NULL	Size of the file in Oracle blocks
STATUS	VARCHAR2(9)		File status: AVAILABLE or INVALID (INVALID means that the file number is not in use, for example, a file in a tablespace that was dropped)
RELATIVE_FNO	NUMBER		Relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator
MAXBYTES	NUMBER		Maximum file size in bytes
MAXBLOCKS	NUMBER		Maximum file size in blocks
INCREMENT_BY	NUMBER		Autoextension increment
USER_BYTES	NUMBER		Corresponding number of bytes
USER_BLOCKS	NUMBER		Number of blocks which can be used by the data

DBA_DATAPUMP_JOBS

DBA_DATAPUMP_JOBS displays all Data Pump jobs in the database.

Related View

USER_DATAPUMP_JOBS displays the Data Pump jobs owned by the current user. This view does not display the OWNER_NAME column.

Column	Datatype	NULL	Description
OWNER_NAME	VARCHAR2(30)		User that initiated the job
JOB_NAME	VARCHAR2(30)		Name of the job
OPERATION	VARCHAR2(30)		Type of operation being performed
JOB_MODE	VARCHAR2(30)		Mode of operation being performed
STATE	VARCHAR2(30)		Current job state
DEGREE	NUMBER		Number of worker processes performing the operation
ATTACHED_SESSIONS	NUMBER		Number of sessions attached to the job

See Also: ["USER_DATAPUMP_JOBS"](#) on page 3-163

DBA_DATAPUMP_SESSIONS

DBA_DATAPUMP_SESSIONS displays all Data pump sessions attached to a job in the database.

Column	Datatype	NULL	Description
OWNER_NAME	VARCHAR2(30)		User that initiated the job
JOB_NAME	VARCHAR2(30)		Name of the job
SADDR	RAW(4)		Address of the session attached to the job

DBA_DB_LINKS

DBA_DB_LINKS describes all database links in the database. Its columns (except for PASSWORD) are the same as those in ALL_DB_LINKS.

See Also: ["ALL_DB_LINKS"](#) on page 2-26

DBA_DDL_LOCKS

DBA_DDL_LOCKS lists all DDL locks held in the database and all outstanding requests for a DDL lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session identifier
OWNER	VARCHAR2(30)		Owner of the lock
NAME	VARCHAR2(30)		Name of the lock

Column	Datatype	NULL	Description
TYPE	VARCHAR2(40)		Lock type: <ul style="list-style-type: none"> ▪ Cursor ▪ Table/Procedure/Type ▪ Body ▪ Trigger ▪ Index ▪ Cluster ▪ Java Source ▪ Java Resource ▪ Java Data
MODE_HELD	VARCHAR2(9)		Lock mode: <ul style="list-style-type: none"> ▪ None ▪ Null ▪ Share ▪ Exclusive
MODE_REQUESTED	VARCHAR2(9)		Lock request type: <ul style="list-style-type: none"> ▪ None ▪ Null ▪ Share ▪ Exclusive

DBA_DEPENDENCIES

DBA_DEPENDENCIES describes all dependencies in the database between procedures, packages, functions, package bodies, and triggers, including dependencies on views created without any database links. Its columns are the same as those in "[ALL_DEPENDENCIES](#)" on page 2-28.

DBA_DIM_ATTRIBUTES

DBA_DIM_ATTRIBUTES describes the relationships between dimension levels and functionally dependent columns in the database. The level columns and the dependent column must be in the same table. This view's columns are the same as those in "[ALL_DIM_ATTRIBUTES](#)" on page 2-28.

DBA_DIM_CHILD_OF

DBA_DIM_CHILD_OF describes a hierarchical relationship of 1 to n between all the pairs of levels in the dimensions in the database. Its columns are the same as those in "[ALL_DIM_CHILD_OF](#)" on page 2-29.

DBA_DIM_HIERARCHIES

DBA_DIM_HIERARCHIES describes all the dimension hierarchies in the database. Its columns are the same as those in "[ALL_DIM_HIERARCHIES](#)" on page 2-29.

DBA_DIM_JOIN_KEY

DBA_DIM_JOIN_KEY describes all joins in the database between two dimension tables. The join is always specified between a parent dimension level column and a

child column. This view's columns are the same as those in "[ALL_DIM_JOIN_KEY](#)" on page 2-29.

DBA_DIM_LEVEL_KEY

DBA_DIM_LEVEL_KEY describes the columns of all dimension levels in the database. This view's columns are the same as those in "[ALL_DIM_LEVEL_KEY](#)" on page 2-30.

DBA_DIM_LEVELS

DBA_DIM_LEVELS describes all dimension levels in the database. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "[ALL_DIM_LEVELS](#)" on page 2-30.

DBA_DIMENSIONS

DBA_DIMENSIONS represents dimension objects. Its columns are the same as those in "[ALL_DIMENSIONS](#)" on page 2-31.

DBA_DIRECTORIES

DBA_DIRECTORIES describes all directory objects in the database. Its columns are the same as those in "[ALL_DIRECTORIES](#)" on page 2-31.

DBA_DML_LOCKS

DBA_DML_LOCKS lists all DML locks held in the database and all outstanding requests for a DML lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
OWNER	VARCHAR2(30)	NOT NULL	Owner of the lock
NAME	VARCHAR2(30)	NOT NULL	Name of the lock
MODE_HELD	VARCHAR2(13)		The type of lock held. The values are: <ul style="list-style-type: none"> ▪ ROWS_S (SS): row share lock ▪ ROW-X (SX): row exclusive lock ▪ SHARE (S): share lock ▪ S/ROW-X (SSX): exclusive lock ▪ NONE: lock requested but not yet obtained
MODE_REQUESTED	VARCHAR2(13)		Lock request type. The values are: <ul style="list-style-type: none"> ▪ ROWS_S (SS): row share lock ▪ ROW-X (SX): row exclusive lock ▪ SHARE (S): share lock ▪ S/ROW-X (SSX): exclusive lock ▪ NONE: Lock identifier obtained; lock not held or requested
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Blocking others

DBA_DMT_FREE_SPACE

DBA_DMT_FREE_SPACE describes the free extents in all dictionary managed tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_ID	NUMBER	NOT NULL	Identifier number of the tablespace containing the extent
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent
BLOCKS	NUMBER	NOT NULL	Size of the extent (in Oracle blocks)

DBA_DMT_USED_EXTENTS

DBA_DMT_USED_EXTENTS describes the extents comprising the segments in all dictionary managed tablespaces in the database.

Column	Datatype	NULL	Description
SEGMENT_FILEID	NUMBER	NOT NULL	File number of the segment header of the extent
SEGMENT_BLOCK	NUMBER	NOT NULL	Block number of the segment header of the extent
TABLESPACE_ID	NUMBER	NOT NULL	Identifier number of the tablespace containing the extent
EXTENT_ID	NUMBER	NOT NULL	Extent number in the segment
FILEID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK	NUMBER	NOT NULL	Starting block number of the extent
LENGTH	NUMBER	NOT NULL	Number of blocks in the extent

DBA_ENABLED_AGGREGATIONS

DBA_ENABLED_AGGREGATIONS displays information about enabled on-demand statistic aggregation.

Column	Datatype	NULL	Description
AGGREGATION_TYPE	VARCHAR2(21)		Type of the aggregation: <ul style="list-style-type: none"> ■ CLIENT_ID ■ SERVICE ■ SERVICE_MODULE ■ SERVICE_MODULE_ACTION
PRIMARY_ID	VARCHAR2(64)		Primary qualifier (specific client identifier or service name)
QUALIFIER_ID1	VARCHAR2(48)		Secondary qualifier (specific module name)
QUALIFIER_ID2	VARCHAR2(32)		Additional qualifier (specific action name)

DBA_ENABLED_TRACES

DBA_ENABLED_TRACES displays information about enabled SQL traces.

Column	Datatype	NULL	Description
TRACE_TYPE	VARCHAR2(21)		Type of the trace: <ul style="list-style-type: none"> ▪ CLIENT_ID ▪ SERVICE ▪ SERVICE_MODULE ▪ SERVICE_MODULE_ACTION
PRIMARY_ID	VARCHAR2(64)		Primary qualifier (specific client identifier or service name)
QUALIFIER_ID1	VARCHAR2(48)		Secondary qualifier (specific module name)
QUALIFIER_ID2	VARCHAR2(32)		Additional qualifier (specific action name)
WAITS	VARCHAR2(5)		Indicates whether waits are traced (TRUE) or not (FALSE)
BINDS	VARCHAR2(5)		Indicates whether binds are traced (TRUE) or not (FALSE)
INSTANCE_NAME	VARCHAR2(16)		Instance name for tracing restricted to named instances

DBA_ERRORS

DBA_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) in the database. Its columns are the same as those in ["ALL_ERRORS"](#) on page 2-31.

DBA_EVALUATION_CONTEXT_TABLES

DBA_EVALUATION_CONTEXT_TABLES describes the tables in all rule evaluation contexts in the database. Its columns are the same as those in [ALL_EVALUATION_CONTEXT_TABLES](#).

See Also: ["ALL_EVALUATION_CONTEXT_TABLES"](#) on page 2-32

DBA_EVALUATION_CONTEXT_VARS

DBA_EVALUATION_CONTEXT_VARS describes the variables in all rule evaluation contexts in the database. Its columns are the same as those in [ALL_EVALUATION_CONTEXT_VARS](#).

See Also: ["ALL_EVALUATION_CONTEXT_VARS"](#) on page 2-33

DBA_EVALUATION_CONTEXTS

DBA_EVALUATION_CONTEXTS describes all rule evaluation contexts in the database. Its columns are the same as those in [ALL_EVALUATION_CONTEXTS](#).

See Also: ["ALL_EVALUATION_CONTEXTS"](#) on page 2-33

DBA_EXP_FILES

DBA_EXP_FILES describes export files.

Column	Datatype	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the export session
EXP_TYPE	VARCHAR2(11)		Type of export file: complete, cumulative, or incremental

DBA_EXP_OBJECTS

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(100)	NOT NULL	Name of the export file
USER_NAME	VARCHAR2(30)	NOT NULL	Name of user who executed export
TIMESTAMP	DATE	NOT NULL	Timestamp of the export session

DBA_EXP_OBJECTS

DBA_EXP_OBJECTS describes objects that have been incrementally exported.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of exported object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of exported object
OBJECT_TYPE	VARCHAR2(12)		Type of exported object
CUMULATIVE	DATE		Timestamp of last cumulative export
INCREMENTAL	DATE	NOT NULL	Timestamp of last incremental export
EXPORT_VERSION	NUMBER(3)	NOT NULL	The ID of the export session

DBA_EXP_VERSION

DBA_EXP_VERSION displays the version number of the last export session.

Column	Datatype	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the last export session

DBA_EXTENTS

DBA_EXTENTS describes the extents comprising the segments in all tablespaces in the database.

Related View

USER_EXTENTS describes the extents comprising the segments owned by the current user's objects. This view does not display the OWNER, FILE_ID, BLOCK_ID, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the segment associated with the extent
SEGMENT_NAME	VARCHAR2(81)		Name of the segment associated with the extent
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(18)		Type of the segment: INDEX PARTITION, TABLE PARTITION
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent
EXTENT_ID	NUMBER		Extent number in the segment
FILE_ID	NUMBER		File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Starting block number of the extent
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER		Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER		Relative file number of the first extent block

See Also: ["USER_EXTENTS"](#) on page 3-164

DBA_EXTERNAL_LOCATIONS

DBA_EXTERNAL_TABLES describes the locations (data sources) of all external tables in the database. Its columns are the same as those in ALL_EXTERNAL_LOCATIONS.

See Also: ["ALL_EXTERNAL_LOCATIONS"](#) on page 2-34

DBA_EXTERNAL_TABLES

DBA_EXTERNAL_TABLES describes all external tables in the database. Its columns are the same as those in ALL_EXTERNAL_TABLES.

See Also: ["ALL_EXTERNAL_TABLES"](#) on page 2-34

DBA_FEATURE_USAGE_STATISTICS

DBA_FEATURE_USAGE_STATISTICS displays information about database feature usage statistics.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database identifier of the database being tracked
NAME	VARCHAR2 (64)	NOT NULL	Name of the feature (See Table 3-1)
VERSION	VARCHAR2 (17)	NOT NULL	Database version in which the feature was tracked
DETECTED_USAGES	NUMBER	NOT NULL	Number of times the system has detected usage for the feature
TOTAL_SAMPLES	NUMBER	NOT NULL	Number of times the system has woken up and checked for feature usage
CURRENTLY_USED	VARCHAR2 (5)		Indicates whether usage was detected the last time the system checked (TRUE) or not (FALSE)
FIRST_USAGE_DATE	DATE		First sample time the system detected usage of the feature
LAST_USAGE_DATE	DATE		Last sample time the system detected usage of the feature
AUX_COUNT	NUMBER		This column stores feature-specific usage data in number format.
FEATURE_INFO	CLOB		This column stores feature-specific usage data in character format.
LAST_SAMPLE_DATE	DATE		Amount of time (in seconds) between the last two usage sample times
LAST_SAMPLE_PERIOD	NUMBER		Amount of time (in hours) between the last two usage sample times
SAMPLE_INTERVAL	NUMBER		Sample interval
DESCRIPTION	VARCHAR2 (128)		Description of the feature and usage detection logic (See Table 3-1)

Table 3-1 DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Advanced Replication	Advanced Replication has been enabled.
Advanced Security	External Global users are configured.
Audit Options	Audit options in use.

Table 3–1 (Cont.) DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Automatic Database Diagnostic Monitor	A task for the Automatic Database Diagnostic Monitor has been executed.
Automatic Segment Space Management (system)	Extents of locally managed tablespaces are managed automatically by Oracle.
Automatic Segment Space Management (user)	Extents of locally managed user tablespaces are managed automatically by Oracle.
Automatic SQL Execution Memory	Sizing of work areas for all dedicated sessions (PGA) is automatic.
Automatic Storage Manager	Automatic Storage Management has been enabled
Automatic Undo Management	Oracle automatically manages undo data using an UNDO tablespace.
Automatic Workload Repository	A manual Automatic Workload Repository (AWR) snapshot was taken in the last sample period.
Change-Aware Incremental Backup	Track blocks that have changed in the database.
Client Identifier	Application User Proxy Authentication: Client Identifier is used at this specific time.
Data Guard	Data Guard, a set of services, is being used to create, maintain, manage, and monitor one or more standby databases.
Data Guard Broker	Data Guard Broker, the framework that handles the creation maintenance, and monitoring of Data Guard, is being used
Data Mining	Oracle Data Mining option is being used.
Dynamic SGA	The Oracle SGA has been dynamically resized through an ALTER SYSTEM SET statement.
File Mapping	File Mapping, the mechanism that shows a complete mapping of a file to logical volumes and physical devices, is being used.
Flashback Database	Flashback Database, a rewind button for the database, is enabled
Internode Parallel Execution	Internode Parallel Execution is being used.
Label Security	Oracle Label Security, that enables label-based access control Oracle applications, is being used.
Locally Managed Tablespaces (system)	There exists tablespaces that are locally managed in the database.
Locally Managed Tablespaces (user)	There exists user tablespaces that are locally managed in the database.
Messaging Gateway	Messaging Gateway, that enables communication between non-Oracle messaging systems and Advanced Queuing (AQ), link configured.
MTTR Advisor	Mean Time to Recover Advisor is enabled.
Multiple Block Sizes	Multiple Block Sizes are being used with this database.
OLAP - Analytic Workspaces	OLAP - the analytic workspaces stored in the database.
OLAP - Cubes	OLAP - number of cubes in the OLAP catalog that are fully mapped and accessible by the OLAP API.
Oracle Managed Files	Database files are being managed by Oracle.
Parallel SQL DDL Execution	Parallel SQL DDL Execution is being used.
Parallel SQL DML Execution	Parallel SQL DML Execution is being used.
Parallel SQL Query Execution	Parallel SQL Query Execution is being used.
Partitioning (system)	Oracle Partitioning option is being used - there is at least one partitioned object created.
Partitioning (user)	Oracle Partitioning option is being used - there is at least one user partitioned object created.
PL/SQL Native Compilation	PL/SQL Native Compilation is being used - there is at least one natively compiled PL/SQL library unit in the database.
Protection Mode - Maximum Availability	Data Guard configuration data protection mode is Maximum Availability.
Protection Mode - Maximum Performance	Data Guard configuration data protection mode is Maximum Performance.

Table 3–1 (Cont.) DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Protection Mode - Maximum Protection	Data Guard configuration data protection mode is Maximum Protection.
Protection Mode - Unprotected	Data Guard configuration data protection mode is Unprotected.
Real Application Clusters (RAC)	Real Application Clusters (RAC) is configured.
Recovery Area	The recovery area is configured.
Recovery Manager (RMAN)	Recovery Manager (RMAN) is being used to backup the database.
RMAN - Disk Backup	Recovery Manager (RMAN) is being used to backup the database to disk.
RMAN - Tape Backup	Recovery Manager (RMAN) is being used to backup the database to tape.
Resource Manager	Oracle Database Resource Manager is being used to control database resources.
Segment Advisor	A task for Segment Advisor has been executed.
Server Parameter File	The server parameter file (SPFILE) was used to startup the database.
SGA/Memory Advisor	SGA/Memory Advisor has been used.
Shared Server	The database is configured as Shared Server, where the server process can service multiple user processes.
Spatial	There is at least one usage of the Oracle Spatial index metadata table.
SQL Access Advisor	A task for SQL Access Advisor has been executed.
SQL Tuning Advisor	SQL Tuning Advisor has been used.
SQL Tuning Set	A SQL Tuning Set has been created in the database.
Standby Archival - LGWR	Data Guard configuration: Remote archival is done by LGWR.
Standby Archival - ARCH	Data Guard configuration: Remote archival is done by ARCH.
Standby Transmission	Standby database: Network transmission mode was chosen for a standby destination.
Streams (system)	Oracle Streams has been configured
Streams (user)	Users have configured Oracle Streams
Tablespace Advisor	Tablespace Advisor has been used.
Transparent Gateway	Heterogeneous Connectivity, access to a non-Oracle system, has been configured.
Undo Advisor	Undo Advisor has been used.
Virtual Private Database (VPD)	Virtual Private Database (VPD) policies are being used.

DBA_FGA_AUDIT_TRAIL

DBA_FGA_AUDIT_TRAIL displays all audit records for fine-grained auditing.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER	NOT NULL	Session id of the query
TIMESTAMP	DATE		Date and time of the query in the local database session time zone
DB_USER	VARCHAR2(30)		Database username who executed the query
OS_USER	VARCHAR2(255)		Operating system username who executed the query
USERHOST	VARCHAR2(128)		Client host machine name
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
EXT_NAME	VARCHAR2(4000)		External name
OBJECT_SCHEMA	VARCHAR2(30)		Owner of the table or view
OBJECT_NAME	VARCHAR2(128)		Name of the table or view
POLICY_NAME	VARCHAR2(30)		Name of the Fine-Grained Auditing Policy

Column	Datatype	NULL	Description
SCN	NUMBER		System change number (SCN) of the query
SQL_TEXT	NVARCHAR2 (2000)		SQL text of the query
SQL_BIND	NVARCHAR2 (2000)		Bind variable data of the query
COMMENT\$TEXT	VARCHAR2 (4000)		Comments
STATEMENT_TYPE	VARCHAR2 (7)		Statement type of the query: <ul style="list-style-type: none"> ■ SELECT ■ INSERT ■ UPDATE ■ DELETE
EXTENDED_TIMESTAMP	TIMESTAMP (6) WITH TIME ZONE		Timestamp of the query in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2 (32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2 (16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW (8)		Transaction identifier of the transaction in which the object is accessed or modified
STATEMENTID	NUMBER		Numeric ID for each statement run (a statement may cause many actions)
ENTRYID	NUMBER		Numeric ID for each audit trail entry in the session

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

DBA_FREE_SPACE

DBA_FREE_SPACE describes the free extents in all tablespaces in the database.

Related View

USER_FREE_SPACE describes the free extents in the tablespaces accessible to the current user.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2 (30)		Name of the tablespace containing the extent
FILE_ID	NUMBER		File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Starting block number of the extent
BYTES	NUMBER		Size of the extent (in bytes)
BLOCKS	NUMBER		Size of the extent (in Oracle blocks)
RELATIVE_FNO	NUMBER		Relative file number of the file containing the extent

See Also: ["USER_FREE_SPACE"](#) on page 3-165

DBA_FREE_SPACE_COALESCED

DBA_FREE_SPACE_COALESCED describes statistics on coalesced space in all tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2 (30)		Name of the tablespace
TOTAL_EXTENTS	NUMBER		Total number of free extents in the tablespace
EXTENTS_COALESCED	NUMBER		Total number of coalesced free extents in the tablespace
PERCENT_EXTENTS_COALESCED	NUMBER		Percentage of coalesced free extents in the tablespace
TOTAL_BYTES	NUMBER		Total number of free bytes in the tablespace
BYTES_COALESCED	NUMBER		Total number of coalesced free bytes in the tablespace
TOTAL_BLOCKS	NUMBER		Total number of free Oracle blocks in the tablespace
BLOCKS_COALESCED	NUMBER		Total number of coalesced free Oracle blocks in the tablespace
PERCENT_BLOCKS_COALESCED	NUMBER		Percentage of coalesced free Oracle blocks in the tablespace

DBA_HIGH_WATER_MARK_STATISTICS

DBA_HIGH_WATER_MARK_STATISTICS displays information about database high-watermark statistics.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Identifier of the database for which the high-watermark statistics are tracked
NAME	VARCHAR2 (64)	NOT NULL	Name of the high-watermark statistic (See Table 3-2)
VERSION	VARCHAR2 (17)	NOT NULL	Database version in which the high-watermarks were tracked
HIGHWATER	NUMBER		Highest value of the statistic seen at sampling time
LAST_VALUE	NUMBER		Value of the statistic at the last sample time
DESCRIPTION	VARCHAR2 (128)		Description of the high-watermark statistic (See Table 3-2)

Table 3-2 DBA_HIGH_WATER_MARK_STATISTICS Statistics

Name	Description
USER_TABLES	Number of User Tables
SEGMENT_SIZE	Size of Largest Segment (Bytes)
PART_TABLES	Maximum Number of Partitions belonging to an User Table
PART_INDEXES	Maximum Number of Partitions belonging to an User Index
USER_INDEXES	Number of User Indexes
SESSIONS	Maximum Number of Concurrent Sessions seen in the database
DB_SIZE	Maximum Size of the Database (Bytes)
DATAFILES	Maximum Number of Datafiles
TABLESPACES	Maximum Number of Tablespaces

Table 3–2 (Cont.) DBA_HIGH_WATER_MARK_STATISTICS Statistics

Name	Description
CPU_COUNT	Maximum Number of CPUs
QUERY_LENGTH	Maximum Query Length
SERVICES	Maximum Number of Services

DBA_HIST_ACTIVE_SESS_HISTORY

DBA_HIST_ACTIVE_SESS_HISTORY displays the history of the contents of the in-memory active session history of recent system activity. This view contains snapshots of V\$ACTIVE_SESSION_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SAMPLE_ID	NUMBER		ID of the sample
SAMPLE_TIME	TIMESTAMP (3)		Time of the sample
SESSION_ID	NUMBER		Session identifier
SESSION_SERIAL#	NUMBER		Session serial number (used to uniquely identify a session's objects)
USER_ID	NUMBER		Oracle user identifier
SQL_ID	VARCHAR2 (13)		SQL identifier of the SQL statement that is currently being executed
SQL_CHILD_NUMBER	NUMBER		Child number of the SQL statement that is currently being executed
SQL_PLAN_HASH_VALUE	NUMBER		Numerical representation of the SQL plan for the cursor
SERVICE_HASH	NUMBER		Hash that identifies the Service
SESSION_TYPE	NUMBER		Session type
SQL_OPCODE	NUMBER		Indicates what phase of operation the SQL statement is in
QC_SESSION_ID	NUMBER		Query coordinator session ID
QC_INSTANCE_ID	NUMBER		Query coordinator instance ID
CURRENT_OBJ#	NUMBER		Object ID of the object that the session is currently referencing
CURRENT_FILE#	NUMBER		File number of the file containing the block that the session is currently referencing
CURRENT_BLOCK#	NUMBER		ID of the block that the session is currently referencing
SEQ#	NUMBER		Sequence number that uniquely identifies the wait (incremented for each wait)
EVENT_ID	NUMBER		Identifier of the resource or event for which the session is waiting or for which the session last waited
P1	NUMBER		First additional parameter
P2	NUMBER		Second additional parameter
P3	NUMBER		Third additional parameter
WAIT_TIME	NUMBER		Total wait time for the event for which the session last waited (0 if currently waiting)
TIME_WAITED	NUMBER		Time that the current session actually spent waiting for the event. This column is set for waits that were in progress at the time the sample was taken.

Column	Datatype	NULL	Description
PROGRAM	VARCHAR2 (48)		Name of the operating system program
MODULE	VARCHAR2 (48)		Name of the currently executing module as set by the DBMS_APPLICATION_INFO.SET_MODULE procedure
ACTION	VARCHAR2 (32)		Name of the currently executing action as set by the DBMS_APPLICATION_INFO.SET_ACTION procedure
CLIENT_ID	VARCHAR2 (64)		Client identifier of the session

DBA_HIST_BASELINE

DBA_HIST_BASELINE displays information on baselines taken in the system.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
BASELINE_ID	NUMBER	NOT NULL	Unique baseline ID
BASELINE_NAME	VARCHAR2 (64)		User-specified name for the baseline
START_SNAP_ID	NUMBER	NOT NULL	Start snapshot ID
START_SNAP_TIME	TIMESTAMP (3)	NOT NULL	Time associated with the start snapshot ID
END_SNAP_ID	NUMBER	NOT NULL	End snapshot ID
END_SNAP_TIME	TIMESTAMP (3)	NOT NULL	Time associated with the end snapshot ID

DBA_HIST_BG_EVENT_SUMMARY

DBA_HIST_BG_EVENT_SUMMARY displays the historical summary background event activity. This view contains snapshots from V\$SESSION_EVENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2 (64)	NOT NULL	Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2 (64)		Name of the class of the wait event
TOTAL_WAITS	NUMBER		Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER		Total number of timeouts for the event
TIME_WAITED_MICRO	NUMBER		Total amount of time waited for the event (in microseconds)

DBA_HIST_BUFFER_POOL_STAT

DBA_HIST_BUFFER_POOL_STAT displays historical statistics about all buffer pools available for the instance. This view contains snapshots of V\$BUFFER_POOL_STATISTICS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot

DBA_HIST_CR_BLOCK_SERVER

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
ID	NUMBER	NOT NULL	Buffer pool identifier number
NAME	VARCHAR2(20)		Name of the buffer pool
BLOCK_SIZE	NUMBER		Block Size
SET_MSIZE	NUMBER		Buffer pool maximum set size
CNUM_REPL	NUMBER		Number of buffers on the replacement list
CNUM_WRITE	NUMBER		Number of buffers on the write list
CNUM_SET	NUMBER		Number of buffers in the set
BUF_GOT	NUMBER		Number of buffers gotten by the set
SUM_WRITE	NUMBER		Number of buffers written by the set
SUM_SCAN	NUMBER		Number of buffers scanned in the set
FREE_BUFFER_WAIT	NUMBER		Free buffer wait statistic
WRITE_COMPLETE_WAIT	NUMBER		Write complete wait statistic
BUFFER_BUSY_WAIT	NUMBER		Buffer busy wait statistic
FREE_BUFFER_INSPECTED	NUMBER		Free buffer inspected statistic
DIRTY_BUFFERS_INSPECTED	NUMBER		Dirty buffers inspected statistic
DB_BLOCK_CHANGE	NUMBER		Database blocks changed statistic
DB_BLOCK_GETS	NUMBER		Database blocks gotten statistic
CONSISTENT_GETS	NUMBER		Consistent gets statistic
PHYSICAL_READS	NUMBER		Physical reads statistic
PHYSICAL_WRITES	NUMBER		Physical writes statistic

DBA_HIST_CR_BLOCK_SERVER

DBA_HIST_CR_BLOCK_SERVER displays historical statistics on the Global Cache Service processes (LMS) used in cache fusion. This view contains snapshots of V\$CR_BLOCK_SERVER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
CR_REQUESTS	NUMBER		Number of CR blocks served due to remote CR block requests
CURRENT_REQUESTS	NUMBER		Number of current blocks served due to remote CR block requests CR_REQUESTS + CURRENT_REQUESTS = global cache cr clocks served (from V\$SYSSTAT).
DATA_REQUESTS	NUMBER		Number of current or CR requests for data blocks
UNDO_REQUESTS	NUMBER		Number of CR requests for undo blocks
TX_REQUESTS	NUMBER		Number of CR requests for undo segment header blocks DATA_REQUESTS + UNDO_REQUESTS + TX_REQUESTS = total number of requests handled by the LMS processes
CURRENT_RESULTS	NUMBER		Number of requests for which no changes were rolled out of the block returned to the requesting instance

Column	Datatype	NULL	Description
PRIVATE_RESULTS	NUMBER		Number of requests for which changes were rolled out of the block returned to the requesting instance, and only the requesting transaction can use the resulting CR block
ZERO_RESULTS	NUMBER		Number of requests for which changes were rolled out of the block returned to the requesting instance. Only zero-XID transactions can use the block.
DISK_READ_RESULTS	NUMBER		Number of requests for which the requesting instance had to read the requested block from disk
FAIL_RESULTS	NUMBER		Number of requests that failed; the requesting transaction must reissue the request
FAIRNESS_DOWN_CONVERTS	NUMBER		Number of times an instance receiving a request has down-converted an X lock on a block because it was not modifying the block
FAIRNESS_CLEARS	NUMBER		Number of times the fairness counter was cleared. This counter tracks the number of times a block was modified after it was served.
FREE_GC_ELEMENTS	NUMBER		Number of times a request was received from another instance and the X lock had no buffers
FLUSHES	NUMBER		Number of times the log has been flushed by an LMS process
FLUSHES_QUEUED	NUMBER		Number of flushes queued by an LMS process
FLUSH_QUEUE_FULL	NUMBER		Number of times the flush queue was full
FLUSH_MAX_TIME	NUMBER		Maximum time for flush
LIGHT_WORKS	NUMBER		Number of times the light-work rule was evoked. This rule prevents the LMS processes from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the LMS process from completing its response to the CR request.
ERRORS	NUMBER		Number of times an error was signalled by an LMS process

DBA_HIST_CURRENT_BLOCK_SERVER

DBA_HIST_CURRENT_BLOCK_SERVER displays historical statistics on the Global Cache Service processes (LMS) used in cache fusion. This view contains snapshots of V\$CURRENT_BLOCK_SERVER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
PIN1	NUMBER		Pins taking less than 1 millisecond
PIN10	NUMBER		Pins taking 1 to 10 milliseconds
PIN100	NUMBER		Pins taking 10 to 100 milliseconds
PIN1000	NUMBER		Pins taking 100 to 1000 milliseconds
PIN10000	NUMBER		Pins taking 1000 to 10000 milliseconds
FLUSH1	NUMBER		Flushes taking less than 1 millisecond
FLUSH10	NUMBER		Flushes taking 1 to 10 milliseconds
FLUSH100	NUMBER		Flushes taking 10 to 100 milliseconds
FLUSH1000	NUMBER		Flushes taking 100 to 1000 milliseconds

DBA_HIST_DATABASE_INSTANCE

Column	Datatype	NULL	Description
FLUSH10000	NUMBER		Flushes taking 1000 to 10000 milliseconds
WRITE1	NUMBER		Writes taking less than 1 millisecond
WRITE10	NUMBER		Writes taking 1 to 10 milliseconds
WRITE100	NUMBER		Writes taking 10 to 100 milliseconds
WRITE1000	NUMBER		Writes taking 100 to 1000 milliseconds
WRITE10000	NUMBER		Writes taking 1000 to 10000 milliseconds

DBA_HIST_DATABASE_INSTANCE

DBA_HIST_DATABASE_INSTANCE displays the databases and instances in the Workload Repository.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number
STARTUP_TIME	TIMESTAMP(3)	NOT NULL	Startup time of the instance
PARALLEL	VARCHAR2(3)	NOT NULL	Indicates whether the instance is running in a RAC environment (YES) or not (NO)
VERSION	VARCHAR2(17)	NOT NULL	Database version
DB_NAME	VARCHAR2(9)		Name of the database
INSTANCE_NAME	VARCHAR2(16)		Name of the instance
HOST_NAME	VARCHAR2(64)		Name of the host
LAST_ASH_SAMPLE_ID	NUMBER	NOT NULL	Last sample ID for the active session history

DBA_HIST_DATAFILE

DBA_HIST_DATAFILE displays a history of the datafile information from the control file. This view contains snapshots of V\$DATAFILE.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the datafile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the datafile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the datafile

DBA_HIST_DB_CACHE_ADVICE

DBA_HIST_DB_CACHE_ADVICE displays historical predictions of the number of physical reads for the cache size corresponding to each row. This view contains snapshots of V\$DB_CACHE_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
BPID	NUMBER		Buffer Pool identifier (ranges from 1 to 8)
BUFFERS_FOR_ESTIMATE	NUMBER		Cache size for prediction (in terms of buffers)
NAME	VARCHAR2(20)		Buffer pool name
BLOCK_SIZE	NUMBER		Block size in bytes for buffers in the pool (the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, or 32768)
ADVICE_STATUS	VARCHAR2(3)		Status of the advisory: <ul style="list-style-type: none"> ■ ON - Currently running ■ OFF - Disabled (the estimates are historical and calculated when last enabled)
SIZE_FOR_ESTIMATE	NUMBER		Cache size for prediction (in megabytes)
SIZE_FACTOR	NUMBER		Size factor with respect to the current cache size
PHYSICAL_READS	NUMBER		Physical reads for the cache size
BASE_PHYSICAL_READS	NUMBER		Base physical reads for the cache size
ACTUAL_PHYSICAL_READS	NUMBER		Actual physical reads for the cache size

DBA_HIST_DLM_MISC

DBA_HIST_DLM_MISC displays miscellaneous RAC statistics. This view contains snapshots of V\$DLM_MISC.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STATISTIC#	NUMBER		Statistic number
NAME	VARCHAR2(38)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_ENQUEUE_STAT

DBA_HIST_ENQUEUE_STAT displays historical statistics on the number of enqueue (lock) requests for each type of lock. This view contains snapshots of V\$ENQUEUE_STATISTICS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
EQ_TYPE	VARCHAR2(2)	NOT NULL	Type of enqueue requested
REQ_REASON	VARCHAR2(64)	NOT NULL	Reason for the request
TOTAL_REQ#	NUMBER		Total number of enqueue requests or enqueue conversions for the type of enqueue
TOTAL_WAIT#	NUMBER		Total number of times an enqueue request or conversion resulted in a wait
SUCC_REQ#	NUMBER		Number of times an enqueue request or conversion was granted

DBA_HIST_EVENT_NAME

Column	Datatype	NULL	Description
FAILED_REQ#	NUMBER		Number of times an enqueue request or conversion failed
CUM_WAIT_TIME	NUMBER		Total amount of time (in milliseconds) spent waiting for the enqueue or enqueue conversion
EVENT#	NUMBER		Event number

DBA_HIST_EVENT_NAME

DBA_HIST_EVENT_NAME displays information about wait events. This view contains a snapshot of V\$EVENT_NAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2(64)	NOT NULL	Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name of the class of the wait event

DBA_HIST_FILEMETRIC_HISTORY

DBA_HIST_FILEMETRIC_HISTORY displays the history of file metrics collected in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FILEID	NUMBER	NOT NULL	File number
CREATIONTIME	NUMBER	NOT NULL	File creation time
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	???
AVGREADTIME	NUMBER	NOT NULL	Average file read time
AVGWRTIME	NUMBER	NOT NULL	Average file write time
PHYSICALREAD	NUMBER	NOT NULL	Number of physical reads
PHYSICALWRITE	NUMBER	NOT NULL	Number of physical writes
PHYBLKREAD	NUMBER	NOT NULL	Number of physical block reads
PHYBLKWRITE	NUMBER	NOT NULL	Number of physical block writes

DBA_HIST_FILESTATXS

DBA_HIST_FILESTATXS displays information about file read/write statistics. This view contains snapshots of V\$FILESTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID

Column	Datatype	NULL	Description
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
FILE#	NUMBER		File identification number
CREATION_CHANGE#	NUMBER		Change number at which the datafile was created
FILENAME	VARCHAR2(513)		Name of the datafile
TS#	NUMBER		Tablespace number
TSNAME	VARCHAR2(30)		Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the datafile
PHYRDS	NUMBER		Number of physical reads done
PHYWRTS	NUMBER		Number of times DBWR is required to write
SINGLEBLKRDS	NUMBER		Number of single block reads
READTIM	NUMBER		Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if TIMED_STATISTICS is false
WRITETIM	NUMBER		Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if TIMED_STATISTICS is false
SINGLEBLKRDTIM	NUMBER		Cumulative single block read time (in hundredths of a second)
PHYBLKRD	NUMBER		Number of physical blocks read
PHYBLKWRT	NUMBER		Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
WAIT_COUNT	NUMBER		Wait Count
TIME	NUMBER		Wait Time

DBA_HIST_INSTANCE_RECOVERY

DBA_HIST_INSTANCE_RECOVERY displays the historical monitoring of the mechanisms available to the user to limit recovery I/O. This view contains snapshots of V\$INSTANCE_RECOVERY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
RECOVERY_ESTIMATED_IOS	NUMBER		Number of dirty buffers in the buffer cache. In the Standard Edition, this column is always null.
ACTUAL_REDO_BKLS	NUMBER		Current actual number of redo blocks required for recovery
TARGET_REDO_BKLS	NUMBER		Current target number of redo blocks that must be processed for recovery. This value is the minimum value of the following 3 columns, and identifies which of the 3 user-defined limits determines checkpointing.
LOG_FILE_SIZE_REDO_BKLS	NUMBER		Maximum number of redo blocks required to guarantee that a log switch does not occur before the checkpoint completes
LOG_CHKPT_TIMEOUT_REDO_BKLS	NUMBER		Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_TIMEOUT parameter. The value displayed is not meaningful unless LOG_CHECKPOINT_TIMEOUT has been set.

Column	Datatype	NULL	Description
LOG_CHKPT_INTERVAL_ REDO_BKLS	NUMBER		Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_INTERVAL parameter. The value displayed is not meaningful unless LOG_CHECKPOINT_INTERVAL has been set.
FAST_START_IO_TARGET_ REDO_BKLS	NUMBER		This column is obsolete and maintained for backward compatibility. The value of this column is always null.
TARGET_MTTR	NUMBER		Effective MTTR (mean time to recover) target value in seconds. The TARGET_MTTR value is calculated based on the value of the FAST_START_MTTR_TARGET parameter (the TARGET_MTTR value is used internally), and is usually an approximation of the parameter's value. However, if the FAST_START_MTTR_TARGET parameter value is very small (for example, one second), or very large (for example, 3600 seconds), then the calculation will produce a target value dictated by system limitations. In such cases, the TARGET_MTTR value will be the shortest calculated time, or the longest calculated time that recovery is expected to take. If FAST_START_MTTR_TARGET is not specified, then the value of this field is the current estimated MTTR.
ESTIMATED_MTTR	NUMBER		Current estimated mean time to recover (MTTR) based on the number of dirty buffers and log blocks (0 if FAST_START_MTTR_TARGET is not specified). This value tells you how long you can expect recovery to take based on the work the system is doing right now.
CKPT_BLOCK_WRITES	NUMBER		Number of blocks written by checkpoint writes
OPTIMAL_LOGFILE_SIZE	NUMBER		Redo log file size (in megabytes) that is considered optimal based on the current setting of FAST_START_MTTR_TARGET. It is recommended that all online redo logs be configured to be at least this value.
ESTD_CLUSTER_ AVAILABLE_TIME	NUMBER		Estimated time (in seconds) that the cluster would become partially available should the instance fail. This column is only meaningful in a Real Application Clusters environment. In a non-RAC environment, the value of this column is null.
WRITES_MTTR	NUMBER		Number of writes driven by the FAST_START_MTTR_TARGET parameter
WRITES_LOGFILE_SIZE	NUMBER		Number of writes driven by the smallest redo log file size
WRITES_LOG_ CHECKPOINT_SETTINGS	NUMBER		Number of writes driven by the LOG_CHECKPOINT_INTERVAL parameter or the LOG_CHECKPOINT_TIMEOUT parameter
WRITES_OTHER_SETTINGS	NUMBER		Number of writes driven by other reasons (such as the deprecated FAST_START_IO_TARGET parameter)
WRITES_AUTOTUNE	NUMBER		Number of writes due to auto-tune checkpointing
WRITES_FULL_THREAD_ CKPT	NUMBER		Number of writes due to full thread checkpoints

DBA_HIST_JAVA_POOL_ADVICE

DBA_HIST_JAVA_POOL_ADVICE displays historical information about estimated parse time in the Java pool for different pool sizes. This view contains snapshots of V\$JAVA_POOL_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
JAVA_POOL_SIZE_FOR_ESTIMATE	NUMBER	NOT NULL	Java pool size for the estimate (in megabytes)
JAVA_POOL_SIZE_FACTOR	NUMBER		Size factor with respect to the current Java pool size
ESTD_LC_SIZE	NUMBER		Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER		Estimated number of library cache memory objects in the Java pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER		Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a Java pool of the specified size. This is the time that would have been spent in reloading the required objects in the Java pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER		Estimated parse time saved factor with respect to the current Java pool size
ESTD_LC_LOAD_TIME	NUMBER		Estimated elapsed time (in seconds) for parsing in a Java pool of the specified size.
ESTD_LC_LOAD_TIME_FACTOR	NUMBER		Estimated load time factor with respect to the current Java pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER		Estimated number of times a library cache memory object was found in a Java pool of the specified size

DBA_HIST_LATCH

DBA_HIST_LATCH displays historical aggregate latch statistics for both parent and child latches, grouped by latch name. This view contains snapshots of V\$LATCH.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch Hash
LATCH_NAME	VARCHAR2 (64)		Latch name
LEVEL#	NUMBER		Latch level
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times

Column	Datatype	NULL	Description
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LATCH_CHILDREN

DBA_HIST_LATCH_CHILDREN displays historical statistics about child latches. This view includes all columns of DBA_HIST_LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other. This view contains snapshots of V\$LATCH_CHILDREN.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch Hash
LATCH_NAME	VARCHAR2(64)		Latch name
CHILD#	NUMBER		Child latch number (unique only to each parent latch)
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LATCH_MISSES_SUMMARY

DBA_HIST_LATCH_MISSES_SUMMARY displays historical summary statistics about missed attempts to acquire a latch. This view contains snapshots of V\$LATCH_MISSES.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
PARENT_NAME	VARCHAR2(50)	NOT NULL	Latch name of a parent latch
WHERE_IN_CODE	VARCHAR2(64)	NOT NULL	Location that attempted to acquire the latch
NWFAIL_COUNT	NUMBER		Number of times that no-wait acquisition of the latch failed

Column	Datatype	NULL	Description
SLEEP_COUNT	NUMBER		Number of times that acquisition attempts caused sleeps
WTR_SLP_COUNT	NUMBER		Number of times a waiter slept from this loc

DBA_HIST_LATCH_NAME

DBA_HIST_LATCH_NAME displays information about decoded latch names for the latches shown in DBA_HIST_LATCH. This view contains a snapshot of V\$LATCHNAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
LATCH_HASH	NUMBER	NOT NULL	Latch hash
LATCH_NAME	VARCHAR2(64)	NOT NULL	Latch name

DBA_HIST_LATCH_PARENT

DBA_HIST_LATCH_PARENT displays historical statistics about parent latches. This view contains snapshots of V\$LATCH_PARENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch hash
LATCH_NAME	VARCHAR2(64)		Latch name
LEVEL#	NUMBER		Latch level
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LIBRARYCACHE

DBA_HIST_LIBRARYCACHE displays historical statistics about library cache performance and activity. This view contains snapshots of V\$LIBRARYCACHE.

DBA_HIST_LOG

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAMESPACE	VARCHAR2(15)	NOT NULL	Library cache namespace
GETS	NUMBER		Number of times a lock was requested for objects of the namespace
GETHITS	NUMBER		Number of times an object's handle was found in memory
PINS	NUMBER		Number of times a PIN was requested for objects of the namespace
PINHITS	NUMBER		Number of times all of the metadata pieces of the library object were found in memory
RELOADS	NUMBER		Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from disk
INVALIDATIONS	NUMBER		Total number of times objects in the namespace were marked invalid because a dependent object was modified
DLM_LOCK_REQUESTS	NUMBER		Number of GET requests lock instance locks
DLM_PIN_REQUESTS	NUMBER		Number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER		Number of release requests PIN instance locks
DLM_INVALIDATION_REQUESTS	NUMBER		Number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER		Number of invalidation pings received from other instances

DBA_HIST_LOG

DBA_HIST_LOG displays historical log file information from the control file. This view contains snapshots of V\$LOG.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
GROUP#	NUMBER	NOT NULL	Log group number
THREAD#	NUMBER	NOT NULL	Log thread number
SEQUENCE#	NUMBER	NOT NULL	Log sequence number
BYTES	NUMBER		Size of the log (in bytes)
MEMBERS	NUMBER		Number of members in the log group
ARCHIVED	VARCHAR2(3)		Archive status (YES) or NO

Column	Datatype	NULL	Description
STATUS	VARCHAR2 (16)		Log status: <ul style="list-style-type: none"> ■ UNUSED - Online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. ■ CURRENT - Current redo log. This implies that the redo log is active. The redo log could be open or closed. ■ ACTIVE - Log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It may or may not be archived. ■ CLEARING - Log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE statement. After the log is cleared, the status changes to UNUSED. ■ CLEARING_CURRENT - Current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. ■ INACTIVE - Log is no longer needed for instance recovery. It may be in use for media recovery. It may or may not be archived. ■ INVALIDATED - Archived the current redo log without a log switch.
FIRST_CHANGE#	NUMBER		Lowest system change number (SCN) in the log
FIRST_TIME	DATE		Time of the first SCN in the log

DBA_HIST_METRIC_NAME

DBA_HIST_METRIC_NAME describes attributes of the set of RDBMS metrics. This view contains a snapshot of V\$METRICNAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
GROUP_ID	NUMBER	NOT NULL	Metric Group ID
GROUP_NAME	VARCHAR2 (64)		Metric group name
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2 (64)	NOT NULL	Metric name
METRIC_UNIT	VARCHAR2 (64)	NOT NULL	Unit of measurement

DBA_HIST_MTTR_TARGET_ADVICE

DBA_HIST_MTTR_TARGET_ADVICE displays historical predictions of the number of physical I/O requests for the MTTR corresponding to each row. The data also includes a physical I/O factor, which is the ratio of the number of estimated I/O requests to the number of I/O requests actually performed by the current MTTR setting during the measurement interval. This view contains snapshots of V\$MTTR_TARGET_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot

Column	Datatype	NULL	Description
MTTR_TARGET_FOR_ESTIMATE	NUMBER		MTTR setting being simulated (equal to the current MTTR setting if this is the first row of the view)
ADVICE_STATUS	VARCHAR2(5)		Current status of MTTR simulation: <ul style="list-style-type: none"> ■ ON SET ■ READY SET ■ OFF
DIRTY_LIMIT	NUMBER		Dirty buffer limit derived from the MTTR being simulated
ESTD_CACHE_WRITES	NUMBER		Estimated number of cache physical writes under the MTTR
ESTD_CACHE_WRITE_FACTOR	NUMBER		Estimated cache physical write ratio under the MTTR. It is the ratio of the estimated number of cache writes to the number of cache writes under the current MTTR setting.
ESTD_TOTAL_WRITES	NUMBER		Estimated total number of physical writes under the MTTR
ESTD_TOTAL_WRITE_FACTOR	NUMBER		Estimated total physical write ratio under the MTTR. It is the ratio of the estimated total number of physical writes to the total number of physical writes under the current MTTR setting.
ESTD_TOTAL_IOS	NUMBER		Estimated total number of I/O requests under the MTTR
ESTD_TOTAL_IO_FACTOR	NUMBER		Estimated total I/O ratio under the MTTR. It is the ratio of the estimated total number of I/O requests to the total number of I/O requests under the current MTTR setting.

DBA_HIST_OPTIMIZER_ENV

DBA_HIST_OPTIMIZER_ENV displays the optimizer environments that have been captured in the Workload Repository. This view is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
OPTIMIZER_ENV_HASH_VALUE	NUMBER	NOT NULL	Hash value for the optimizer environment
OPTIMIZER_ENV	RAW(623)		Optimizer environment

DBA_HIST_OSSTAT

DBA_HIST_OSSTAT displays historical operating system statistics. This view contains snapshots of V\$OSSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic ID
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_OSSTAT_NAME

DBA_HIST_OSSTAT_NAME displays the names of the operating system statistics. This view is used with DBA_HIST_OSSTAT.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
STAT_ID	NUMBER	NOT NULL	Statistic ID
STAT_NAME	VARCHAR2(64)	NOT NULL	Statistic name

DBA_HIST_PARAMETER

DBA_HIST_PARAMETER displays historical information about the initialization parameters that were in effect for the instance. This view contains snapshots of V\$SYSTEM_PARAMETER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
PARAMETER_HASH	NUMBER		Parameter hash
PARAMETER_NAME	VARCHAR2(64)		Name of the parameter
VALUE	VARCHAR2(512)		Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value
ISDEFAULT	VARCHAR2(9)		Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISMODIFIED	VARCHAR2(10)		Indicates whether the parameter has been modified after instance startup: <ul style="list-style-type: none"> ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup

DBA_HIST_PARAMETER_NAME

DBA_HIST_PARAMETER_NAME displays information about the parameter names captured in the workload repository. This view is used with the DBA_HIST_PARAMETER view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
PARAMETER_HASH	NUMBER	NOT NULL	Parameter hash
PARAMETER_NAME	VARCHAR2(64)	NOT NULL	Name of the parameter

DBA_HIST_PGA_TARGET_ADVICE

DBA_HIST_PGA_TARGET_ADVICE displays historical predictions of how the cache hit percentage and over allocation count statistics displayed by the V\$PGASTAT

performance view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. This view contains snapshots of V\$PGA_TARGET_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
PGA_TARGET_FOR_ESTIMATE	NUMBER	NOT NULL	Value of PGA_AGGREGATE_TARGET for the prediction (in bytes)
PGA_TARGET_FACTOR	NUMBER		PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2(3)		Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
BYTES_PROCESSED	NUMBER		Total bytes processed by all the work areas considered by this advice (in bytes)
ESTD_EXTRA_BYTES_RW	NUMBER		Estimated number of extra bytes which would be read or written if PGA_AGGREGATE_TARGET was set to the value of the PGA_TARGET_FOR_ESTIMATE column. This number is derived from the estimated number and size of work areas which would run in one-pass (or multi-pass) for that value of PGA_AGGREGATE_TARGET.
ESTD_PGA_CACHE_HIT_PERCENTAGE	NUMBER		Estimated value of the cache hit percentage statistic when PGA_AGGREGATE_TARGET equals PGA_TARGET_FOR_ESTIMATE. This column is derived from the above two columns and is equal to BYTES_PROCESSED / (BYTES_PROCESSED + ESTD_EXTRA_BYTES_RW)
ESTD_OVERALLOC_COUNT	NUMBER		Estimated number of PGA memory over-allocations if the value of PGA_AGGREGATE_TARGET is set to PGA_TARGET_FOR_ESTIMATE. A nonzero value means that PGA_TARGET_FOR_ESTIMATE is not large enough to run the work area workload. Hence, PGA_AGGREGATE_TARGET should not be set to PGA_TARGET_FOR_ESTIMATE since Oracle will not be able to honor that target.

DBA_HIST_PGASTAT

DBA_HIST_PGASTAT displays historical PGA memory usage statistics as well as statistics about the automatic PGA memory manager when it is enabled. This view contains snapshots of V\$PGASTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Unique snapshot ID

Column	Datatype	NULL	Description
NAME	VARCHAR2(64)	NOT NULL	Name of the statistic: <ul style="list-style-type: none"> ▪ aggregate PGA target parameter ▪ aggregate PGA auto target ▪ global memory bound ▪ total PGA allocated ▪ total PGA used ▪ total PGA used for auto workareas ▪ total PGA used for manual workareas ▪ over allocation count ▪ bytes processed ▪ extra bytes read/written ▪ cache hit percentage <p>See Also: V\$PGASTAT for descriptions of the statistics</p>
VALUE	NUMBER		Statistic value

DBA_HIST_RESOURCE_LIMIT

DBA_HIST_RESOURCE_LIMIT displays historical information about global resource use for some of the system resource. This view contains snapshots of V\$RESOURCE_LIMIT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
RESOURCE_NAME	VARCHAR2(30)	NOT NULL	Name of the resource
CURRENT_UTILIZATION	NUMBER		Number of (resources, locks, or processes) currently being used
MAX_UTILIZATION	NUMBER		Maximum consumption of the resource since the last instance start up
INITIAL_ALLOCATION	VARCHAR2(10)		Initial allocation. This will be equal to the value specified for the resource in the initialization parameter file (UNLIMITED for infinite allocation).
LIMIT_VALUE	VARCHAR2(10)		Unlimited for resources and locks. This can be greater than the initial allocation value (UNLIMITED for infinite limit).

DBA_HIST_ROLLSTAT

DBA_HIST_ROLLSTAT displays historical rollback segment statistics. This view contains snapshots of V\$ROLLSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
USN	NUMBER	NOT NULL	Rollback segment number
EXTENTS	NUMBER		Number of extents in the rollback segment

Column	Datatype	NULL	Description
RSSIZE	NUMBER		Size (in bytes) of the rollback segment. This value differs from the value of the BYTES column of the DBA_SEGMENTS views by the number of bytes in one database block.
WRITES	NUMBER		Number of bytes written to the rollback segment
XACTS	NUMBER		Number of active transactions
GETS	NUMBER		Number of header gets
WAITS	NUMBER		Number of header waits
OPTSIZE	NUMBER		Optimal size of the rollback segment
HWMsize	NUMBER		High-watermark of the rollback segment size
SHRINKS	NUMBER		Number of times the size of a rollback segment decreased
WRAPS	NUMBER		Number of times the rollback segment has wrapped
EXTENDS	NUMBER		Number of times the rollback segment size has extended
AVESHRINK	NUMBER		Average shrink size
AVEACTIVE	NUMBER		Current size of active extents, averaged over time

DBA_HIST_ROWCACHE_SUMMARY

DBA_HIST_ROWCACHE_SUMMARY displays historical summary statistics for data dictionary activity. This view contains snapshots of V\$ROWCACHE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
PARAMETER	VARCHAR2(32)		Name of the initialization parameter that determines the number of entries in the data dictionary cache
TOTAL_USAGE	NUMBER		Sum of the total number of entries in the cache
USAGE	NUMBER		Number of cache entries that contain valid data
GETS	NUMBER		Total number of requests for information on the data object
GETMISSES	NUMBER		Number of data requests resulting in cache misses
SCANS	NUMBER		Number of scan requests
SCANMISSES	NUMBER		Number of times a scan failed to find the data in the cache
SCANCOMPLETES	NUMBER		For a list of subordinate entries, the number of times the list was scanned completely
MODIFICATIONS	NUMBER		Number of inserts, updates, and deletions
FLUSHES	NUMBER		Number of times flushed to disk
DLM_REQUESTS	NUMBER		Number of DLM requests
DLM_CONFLICTS	NUMBER		Number of DLM conflicts
DLM_RELEASES	NUMBER		Number of DLM releases

DBA_HIST_SEG_STAT

DBA_HIST_SEG_STAT displays historical information about segment-level statistics. This view captures the top segments based on a set of criteria and captures

information from V\$SEGSTAT. The total value is the value of the statistics since instance startup. The delta value is the value of the statistics from the BEGIN_INTERVAL_TIME to the END_INTERVAL_TIME in the DBA_HIST_SNAPSHOT view.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
TS#	NUMBER		Tablespace number
OBJ#	NUMBER		Dictionary object number
DATAOBJ#	NUMBER		Data object number
LOGICAL_READS_TOTAL	NUMBER		Cumulative value for logical reads
LOGICAL_READS_DELTA	NUMBER		Delta value for logical reads
BUFFER_BUSY_WAITS_TOTAL	NUMBER		Cumulative value for buffer busy waits
BUFFER_BUSY_WAITS_DELTA	NUMBER		Delta value for buffer busy waits
DB_BLOCK_CHANGES_TOTAL	NUMBER		Cumulative value for db block changes
DB_BLOCK_CHANGES_DELTA	NUMBER		Delta value for db block changes
PHYSICAL_READS_TOTAL	NUMBER		Cumulative value for physical reads
PHYSICAL_READS_DELTA	NUMBER		Delta value for physical reads
PHYSICAL_WRITES_TOTAL	NUMBER		Cumulative value for physical writes
PHYSICAL_WRITES_DELTA	NUMBER		Delta value for physical writes
PHYSICAL_READS_DIRECT_TOTAL	NUMBER		Cumulative value for physical reads direct
PHYSICAL_READS_DIRECT_DELTA	NUMBER		Delta value for physical reads direct
PHYSICAL_WRITES_DIRECT_TOTAL	NUMBER		Cumulative value for physical writes direct
PHYSICAL_WRITES_DIRECT_DELTA	NUMBER		Delta value for physical writes direct
ITL_WAITS_TOTAL	NUMBER		Cumulative value for ITL waits
ITL_WAITS_DELTA	NUMBER		Delta value for ITL waits
ROW_LOCK_WAITS_TOTAL	NUMBER		Cumulative value for row lock waits
ROW_LOCK_WAITS_DELTA	NUMBER		Delta value for row lock waits
GC_CR_BLOCKS_SERVED_TOTAL	NUMBER		Cumulative value for global cache CR blocks served
GC_CR_BLOCKS_SERVED_DELTA	NUMBER		Delta value for global cache CR blocks served
GC_CU_BLOCKS_SERVED_TOTAL	NUMBER		Cumulative value for global cache current blocks served
GC_CU_BLOCKS_SERVED_DELTA	NUMBER		Delta value for global cache current blocks served
SPACE_USED_TOTAL	NUMBER		Cumulative value for space used
SPACE_USED_DELTA	NUMBER		Delta value for space used
SPACE_ALLOCATED_TOTAL	NUMBER		Cumulative value for space allocated
SPACE_ALLOCATED_DELTA	NUMBER		Delta value for space allocated
TABLE_SCANS_TOTAL	NUMBER		Cumulative value for table scans

Column	Datatype	NULL	Description
TABLE_SCANS_DELTA	NUMBER		Delta value for table scans

DBA_HIST_SEG_STAT_OBJ

DBA_HIST_SEG_STAT_OBJ displays all the names of the segments captured in the workload repository. This view is used with the DBA_HIST_SEG_STAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
TS#	NUMBER	NOT NULL	Tablespace number
OBJ#	NUMBER	NOT NULL	Dictionary object number
DATAOBJ#	NUMBER	NOT NULL	Data object number
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(18)		Type of the object
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace Name for the object
PARTITION_TYPE	VARCHAR2(8)		Partition Type, if relevant

DBA_HIST_SERVICE_NAME

DBA_HIST_SERVICE_NAME displays the names of the Services tracked by the Workload Repository. This view contains information for V\$SERVICES.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
SERVICE_NAME_HASH	NUMBER	NOT NULL	Hash of the service name
SERVICE_NAME	VARCHAR2(64)	NOT NULL	Name of the service

DBA_HIST_SERVICE_STAT

DBA_HIST_SERVICE_STAT displays the history of important service statistics tracked by the Workload Repository. The call rate statistics in this view can be used for making run-time routing decisions, for tracking service levels, and for per-instance diagnostics per call rate.

The elapsed timing for each call provides a relative value across instances for how well a node is processing SQL calls issued under a service name. When aggregation is enabled for the service name, this view provides historical data on the timing and work done for calls issued for the whole service. This view contains information from V\$SERVICE_STATS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SERVICE_NAME_HASH	NUMBER		Hash of the service name
SERVICE_NAME	VARCHAR2(64)		Name of the service

Column	Datatype	NULL	Description
STAT_ID	NUMBER		Statistic identifier
STAT_NAME	VARCHAR2 (64)		Statistic name
VALUE	NUMBER		Value of the statistic

DBA_HIST_SERVICE_WAIT_CLASS

DBA_HIST_SERVICE_WAIT_CLASS displays the history of wait class information for services as tracked by the Workload Repository. This view contains information from V\$SERVICE_WAIT_CLASS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SERVICE_NAME_HASH	NUMBER		Hash of the service name
SERVICE_NAME	VARCHAR2 (64)		Name of the service
WAIT_CLASS_ID	NUMBER		Identifier for the class of the wait event
WAIT_CLASS	VARCHAR2 (64)		Name for the class of the wait event
TOTAL_WAITS	NUMBER		Total number of waits for this event
TIME_WAITED	NUMBER		Total amount of time waited for this event (in hundredths of a second)

DBA_HIST_SESSMETRIC_HISTORY

DBA_HIST_SESSMETRIC_HISTORY displays the history of several important session metrics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
SESSID	NUMBER	NOT NULL	Session ID
SERIAL#	NUMBER	NOT NULL	Session serial number
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2 (64)	NOT NULL	Metric name
VALUE	NUMBER	NOT NULL	Metric Value
METRIC_UNIT	VARCHAR2 (64)	NOT NULL	Unit of measurement

DBA_HIST_SGA

DBA_HIST_SGA displays historical summary information about the system global area (SGA). This view contains snapshots of V\$SGA.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAME	VARCHAR2(64)	NOT NULL	SGA component group
VALUE	NUMBER	NOT NULL	Memory size (in bytes)

DBA_HIST_SGASTAT

DBA_HIST_SGASTAT displays detailed historical information on the system global area (SGA). This view contains snapshots of V\$SGASTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
NAME	VARCHAR2(64)		SGA component group
POOL	VARCHAR2(12)		Designates the pool in which the memory in NAME resides: <ul style="list-style-type: none"> ▪ <code>shared pool</code> - Memory is allocated from the shared pool ▪ <code>large pool</code> - Memory is allocated from the large pool ▪ <code>java pool</code> - Memory is allocated from the Java pool ▪ <code>streams pool</code> - Memory is allocated from the Streams pool
BYTES	NUMBER		Memory size (in bytes)

DBA_HIST_SHARED_POOL_ADVICE

DBA_HIST_SHARED_POOL_ADVICE displays historical information about estimated parse time in the shared pool for different pool sizes. This view contains snapshots of V\$SHARED_POOL_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SHARED_POOL_SIZE_FOR_ESTIMATE	NUMBER	NOT NULL	Shared pool size for the estimate (in megabytes)
SHARED_POOL_SIZE_FACTOR	NUMBER		Size factor with respect to the current shared pool size
ESTD_LC_SIZE	NUMBER		Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER		Estimated number of library cache memory objects in the shared pool of the specified size

Column	Datatype	NULL	Description
ESTD_LC_TIME_SAVED	NUMBER		Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a shared pool of the specified size. This is the time that would have been spent in reloading the required objects in the shared pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER		Estimated parse time saved factor with respect to the current shared pool size
ESTD_LC_LOAD_TIME	NUMBER		Estimated elapsed time (in seconds) for parsing in a shared pool of the specified size.
ESTD_LC_LOAD_TIME_FACTOR	NUMBER		Estimated load time factor with respect to the current shared pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER		Estimated number of times a library cache memory object was found in a shared pool of the specified size

DBA_HIST_SNAP_ERROR

DBA_HIST_SNAP_ERROR displays information about the snapshot error information in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table in which the error occurred
ERROR_NUMBER	NUMBER	NOT NULL	Error number for the error encountered

DBA_HIST_SNAPSHOT

DBA_HIST_SNAPSHOT displays information about the snapshots in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
STARTUP_TIME	TIMESTAMP(3)	NOT NULL	Startup time of the instance
BEGIN_INTERVAL_TIME	TIMESTAMP(3)	NOT NULL	Time at the beginning of the snapshot interval
END_INTERVAL_TIME	TIMESTAMP(3)	NOT NULL	Time at the end of the snapshot interval; the actual time the snapshot was taken
FLUSH_ELAPSED	INTERVAL DAY(5) TO SECOND(1)		Amount of time to perform the snapshot
SNAP_LEVEL	NUMBER		Snapshot level
ERROR_COUNT	NUMBER		Number of errors occurring in the tables for the particular snapshot

DBA_HIST_SQL_PLAN

DBA_HIST_SQL_PLAN displays the execution plan information for each child cursor in the workload repository. This view captures information from V\$SQL_PLAN and is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
ID	NUMBER	NOT NULL	A number assigned to each step in the execution plan
OPERATION	VARCHAR2(30)		Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)		A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(40)		Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER		Object number of the table or the index
OBJECT_OWNER	VARCHAR2(30)		Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)		Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)		Alias for the object
OBJECT_TYPE	VARCHAR2(20)		Type of the object
OPTIMIZER	VARCHAR2(20)		Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
PARENT_ID	NUMBER		ID of the next execution step that operates on the output of the current step
DEPTH	NUMBER		Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER		Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER		Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER		Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER		Estimate, by the cost-based optimizer, of the number of bytes produced by the operation
OTHER_TAG	VARCHAR2(35)		Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER		Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)		Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)		Stores the method used to distribute rows from producer query servers to consumer query servers

Column	Datatype	NULL	Description
CPU_COST	NUMBER		CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER		I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER		Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)		Expressions produced by the operation
TIME	NUMBER		Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)		Name of the query block
REMARKS	VARCHAR2(4000)		Remarks

DBA_HIST_SQL_SUMMARY

DBA_HIST_SQL_SUMMARY displays historical SQL summary information.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
TOTAL_SQL	NUMBER	NOT NULL	Total number of SQL
TOTAL_SQL_MEM	NUMBER	NOT NULL	Total memory usage of SQL
SINGLE_USE_SQL	NUMBER	NOT NULL	Number of single use SQL
SINGLE_USE_SQL_MEM	NUMBER	NOT NULL	Total memory usage of single use SQL

DBA_HIST_SQL_WORKAREA_HSTGRM

DBA_HIST_SQL_WORKAREA_HSTGRM displays the historical cumulative work area execution statistics (cumulated since instance startup) for different work area groups. This view contains snapshots of V\$SQL_WORKAREA_HISTOGRAM.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
LOW_OPTIMAL_SIZE	NUMBER	NOT NULL	Lower bound for the optimal memory requirement of work areas included in the row (in bytes)
HIGH_OPTIMAL_SIZE	NUMBER	NOT NULL	Upper bound for the optimal memory requirement of work areas included in the row (in bytes)

DBA_HIST_SQLBIND

Column	Datatype	NULL	Description
OPTIMAL_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in optimal mode since instance startup
ONEPASS_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in one-pass mode since instance startup
MULTIPASSES_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in multi-pass mode since instance startup
TOTAL_EXECUTIONS	NUMBER		Sum of OPTIMAL_EXECUTIONS, ONEPASS_EXECUTIONS, and MULTIPASSES_EXECUTIONS

DBA_HIST_SQLBIND

DBA_HIST_SQLBIND displays historical information on bind variables used by SQL cursors. This view contains snapshots of V\$SQL_BIND_CAPTURE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SQL_ID	VARCHAR2(13)		SQL identifier of the parent cursor in the library cache
CHILD_NUMBER	NUMBER		Child cursor number
NAME	VARCHAR2(30)		Name of the bind variable
POSITION	NUMBER		Position of the bind variable in the SQL statement
DUP_POSITION	NUMBER		If the binding is performed by name and the bind variable is duplicated, then this column gives the position of the primary bind variable.
DATATYPE	NUMBER		Internal identifier for the bind datatype
DATATYPE_STRING	VARCHAR2(15)		Textual representation of the bind datatype
CHARACTER_SID	NUMBER		National character set identifier
PRECISION	NUMBER		Precision (for numeric binds)
SCALE	NUMBER		Scale (for numeric binds)
MAX_LENGTH	NUMBER		Maximum bind length
WAS_CAPTURED	VARCHAR2(3)		Indicates whether the bind value was captured (YES) or not (NO)
LAST_CAPTURED	DATE		Date when the bind value was captured. Bind values are captured when SQL statements are executed. To limit the overhead, binds are captured at most every 15 minutes for a given cursor.
VALUE_STRING	VARCHAR2(4000)		Value of the bind represented as a string
VALUE_ANYDATA	ANYDATA		Value of the bind represented using the self-descriptive Sys.AnyData datatype. This representation is useful to programmatically decode the value of the bind variable.

DBA_HIST_SQLSTAT

DBA_HIST_SQLSTAT displays historical information about SQL statistics. This view captures the top SQL statements based on a set of criteria and captures the statistics

information from V\$SQL. The total value is the value of the statistics since instance startup. The delta value is the value of the statistics from the BEGIN_INTERVAL_TIME to the END_INTERVAL_TIME in the DBA_HIST_SNAPSHOT view.

This view is used with the DBA_HIST_OPTIMIZER_ENV, DBA_HIST_SQLTEXT, and DBA_HIST_SQL_PLAN views to provide a complete picture of historical SQL statistics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SQL_ID	VARCHAR2(13)		SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER		Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
OPTIMIZER_COST	NUMBER		Cost of the query given by the optimizer
OPTIMIZER_MODE	VARCHAR2(10)		Mode under which the SQL statement is executed
OPTIMIZER_ENV_HASH_VALUE	NUMBER		Hash Value for the optimizer environment
SHARABLE_MEM	NUMBER		Amount of shared memory used by the child cursor (in bytes)
LOADED_VERSIONS	NUMBER		Indicates whether the context heap is loaded (1) or not (0)
VERSION_COUNT	NUMBER		Number of children associated with the cursor
MODULE	VARCHAR2(64)		Contains the name of the module that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_MODULE
ACTION	VARCHAR2(64)		Contains the name of the action that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_ACTION
SQL_PROFILE	VARCHAR2(64)		Name of the applied SQL Profile
PARSING_SCHEMA_ID	NUMBER		Schema ID that was used to originally build the child cursor
FETCHES_TOTAL	NUMBER		Cumulative number of fetches associated with the SQL statement
FETCHES_DELTA	NUMBER		Delta number of fetches associated with the SQL statement
END_OF_FETCH_COUNT_TOTAL	NUMBER		Cumulative number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
END_OF_FETCH_COUNT_DELTA	NUMBER		Delta number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed.
SORTS_TOTAL	NUMBER		Cumulative number of sorts that were done for this child cursor

Column	Datatype	NULL	Description
SORTS_DELTA	NUMBER		Delta number of sorts that were done for this child cursor
EXECUTIONS_TOTAL	NUMBER		Cumulative number of executions that took place on this object since it was brought into the library cache
EXECUTIONS_DELTA	NUMBER		Delta number of executions that took place on this object since it was brought into the library cache
LOADS_TOTAL	NUMBER		Cumulative number of times the object was either loaded or reloaded
LOADS_DELTA	NUMBER		Delta number of times the object was either loaded or reloaded
INVALIDATIONS_TOTAL	NUMBER		Cumulative number of times this child cursor has been invalidated
INVALIDATIONS_DELTA	NUMBER		Delta number of times this child cursor has been invalidated
PARSE_CALLS_TOTAL	NUMBER		Cumulative number of parse calls for this child cursor
PARSE_CALLS_DELTA	NUMBER		Delta number of parse calls for this child cursor
DISK_READS_TOTAL	NUMBER		Cumulative number of disk reads for this child cursor
DISK_READS_DELTA	NUMBER		Delta number of disk reads for this child cursor
BUFFER_GETS_TOTAL	NUMBER		Cumulative number of buffer gets for this child cursor
BUFFER_GETS_DELTA	NUMBER		Delta number of buffer gets for this child cursor
ROWS_PROCESSED_TOTAL	NUMBER		Cumulative number of rows the parsed SQL statement returns
ROWS_PROCESSED_DELTA	NUMBER		Delta number of rows the parsed SQL statement returns
CPU_TIME_TOTAL	NUMBER		Cumulative value of CPU time (in microseconds) used by this cursor for parsing/executing/fetching
CPU_TIME_DELTA	NUMBER		Delta value of CPU time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME_TOTAL	NUMBER		Cumulative value of elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME_DELTA	NUMBER		Delta value of elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
IOWAIT_TOTAL	NUMBER		Cumulative value of user I/O wait time
IOWAIT_DELTA	NUMBER		Delta value of user I/O wait time
CLWAIT_TOTAL	NUMBER		Cumulative value of cluster wait time
CLWAIT_DELTA	NUMBER		Delta value of cluster wait time
APWAIT_TOTAL	NUMBER		Cumulative value of application wait time
APWAIT_DELTA	NUMBER		Delta value of application wait time
CCWAIT_TOTAL	NUMBER		Cumulative value of concurrency wait time
CCWAIT_DELTA	NUMBER		Delta value of concurrency wait time
DIRECT_WRITES_TOTAL	NUMBER		Cumulative value of direct writes
DIRECT_WRITES_DELTA	NUMBER		Delta value of direct writes
PLSEXEC_TIME_TOTAL	NUMBER		Cumulative value of PL/SQL Execution Time
PLSEXEC_TIME_DELTA	NUMBER		Delta value of PL/SQL Execution Time
JAVEXEC_TIME_TOTAL	NUMBER		Cumulative value of Java Execution Time
JAVEXEC_TIME_DELTA	NUMBER		Delta value of Java Execution Time

DBA_HIST_SQLTEXT

DBA_HIST_SQLTEXT displays the text of SQL statements belonging to shared SQL cursors captured in the Workload Repository. This view captures information from V\$SQL and is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
SQL_TEXT	CLOB		Full text for the SQL statement exposed as a CLOB column
COMMAND_TYPE	NUMBER		Oracle command type definition

DBA_HIST_STAT_NAME

DBA_HIST_STAT_NAME displays decoded statistic names for the statistics captured in the Workload Repository. This view captures information from V\$STATNAME and used with DBA_HIST_SYSSTAT and DBA_HIST_SYS_TIME_MODEL.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
STAT_ID	NUMBER	NOT NULL	Statistic identifier
STAT_NAME	VARCHAR2(64)	NOT NULL	Statistic name

DBA_HIST_SYS_TIME_MODEL

DBA_HIST_SYS_TIME_MODEL displays historical system time model statistics. This view contains snapshots of V\$SYS_TIME_MODEL.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic ID
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_SYSMETRIC_HISTORY

DBA_HIST_SYSMETRIC_HISTORY externalizes all available history of the system metric values for the entire set of data kept in the database. This view contains snapshots of V\$SYSMETRIC_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval

Column	Datatype	NULL	Description
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
VALUE	NUMBER	NOT NULL	Metric Value
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement

DBA_HIST_SYSMETRIC_SUMMARY

DBA_HIST_SYSMETRIC_SUMMARY displays a history of statistical summary of all metric values in the System Metrics Long Duration group. This view contains snapshots of V\$SYSMETRIC_SUMMARY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement
NUM_INTERVAL	NUMBER	NOT NULL	Number of intervals observed
MINVAL	NUMBER	NOT NULL	Minimum value observed
MAXVAL	NUMBER	NOT NULL	Maximum value observed
AVERAGE	NUMBER	NOT NULL	Average over the period
STANDARD_DEVIATION	NUMBER	NOT NULL	One standard deviation

DBA_HIST_SYSSTAT

DBA_HIST_SYSSTAT displays historical system statistics information. This view contains snapshots of V\$SYSSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic identifier
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_SYSTEM_EVENT

DBA_HIST_SYSTEM_EVENT displays historical information on total waits for an event. This view contains snapshots of V\$SYSTEM_EVENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
EVENT_ID	NUMBER		Identifier of the wait event
EVENT_NAME	VARCHAR2(64)		Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the Class of the Wait Event
WAIT_CLASS	VARCHAR2(64)		Name of the Class of the Wait Event
TOTAL_WAITS	NUMBER		Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER		Total number of timeouts for the event
TIME_WAITED_MICRO	NUMBER		Total amount of time waited for the event (in microseconds)

DBA_HIST_TABLESPACE_STAT

DBA_HIST_TABLESPACE_STAT displays tablespace information from the control file. This view contains snapshots of V\$TABLESPACE and DBA_TABLESPACE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
TS#	NUMBER		Tablespace number
TSNAME	VARCHAR2(30)		Tablespace name
CONTENTS	VARCHAR2(9)		Tablespace contents: <ul style="list-style-type: none"> ■ PERMANENT ■ TEMPORARY
STATUS	VARCHAR2(9)		Tablespace status: <ul style="list-style-type: none"> ■ ONLINE ■ OFFLINE ■ READ ONLY
SEGMENT_SPACE_MANAGEMENT	VARCHAR2(6)		Indicates whether the free and used segment space in the tablespace is managed using free lists (MANUAL) or bitmaps (AUTO)
EXTENT_MANAGEMENT	VARCHAR2(10)		Indicates whether the extents in the tablespace are dictionary managed (DICTIONARY) or locally managed (LOCAL)
IS_BACKUP	VARCHAR2(5)		Indicates whether the tablespace is part of a backup

DBA_HIST_TBSPC_SPACE_USAGE

DBA_HIST_TBSPC_SPACE_USAGE displays historical tablespace usage statistics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
TABLESPACE_ID	NUMBER		Tablespace ID
TABLESPACE_SIZE	NUMBER		Tablespace Size
TABLESPACE_MAXSIZE	NUMBER		Maximum size of the tablespace
TABLESPACE_USEDSIZE	NUMBER		Used size of the tablespace
RTIME	VARCHAR2(25)		???

DBA_HIST_TEMPFILE

DBA_HIST_TEMPFILE displays a history of the tempfile information from the control file. This view contains snapshots of V\$TEMPFILE.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the tempfile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the tempfile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the tempfile

DBA_HIST_TEMPSTATXS

DBA_HIST_TEMPSTATXS displays information about temporary file read/write statistics. This view contains snapshots of V\$TEMPSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the tempfile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the tempfile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the tempfile
PHYRDS	NUMBER		Number of physical reads done
PHYWRTS	NUMBER		Number of times DBWR is required to write
SINGLEBLKRDS	NUMBER		Number of single block reads
READTIM	NUMBER		Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER		Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER		Cumulative single block read time (in hundredths of a second)
PHYBLKRD	NUMBER		Number of physical blocks read

Column	Datatype	NULL	Description
PHYBLKWRT	NUMBER		Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
WAIT_COUNT	NUMBER		Wait Count
TIME	NUMBER		Wait time

DBA_HIST_THREAD

DBA_HIST_THREAD displays historical thread information from the control file.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
THREAD#	NUMBER	NOT NULL	Thread number
THREAD_INSTANCE_NUMBER	NUMBER		Instance number of the thread
STATUS	VARCHAR2 (6)		Thread status (OPEN) or (CLOSED)
OPEN_TIME	DATE		Last time the thread was opened
CURRENT_GROUP#	NUMBER		Current log group
SEQUENCE#	NUMBER		Sequence number of the current log

DBA_HIST_UNDOSTAT

DBA_HIST_UNDOSTAT displays the history of histograms of statistical data to show how well the system is working. The available statistics include undo space consumption, transaction concurrency, and length of queries executed in the instance. This view contains snapshots of V\$UNDOSTAT.

Column	Datatype	NULL	Description
BEGIN_TIME	DATE	NOT NULL	Identifies the beginning of the time interval
END_TIME	DATE	NOT NULL	Identifies the end of the time interval
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
UNDOTSN	NUMBER	NOT NULL	Represents the last active undo tablespace in the duration of time. The tablespace ID of the active undo tablespace is returned in this column. If more than one undo tablespace was active in that period, the active undo tablespace that was active at the end of the period is reported.
UNDOBLKS	NUMBER		Represents the total number of undo blocks consumed. You can use this column to obtain the consumption rate of undo blocks, and thereby estimate the size of the undo tablespace needed to handle the workload on your system.
TXNCOUNT	NUMBER		Identifies the total number of transactions executed within the period

Column	Datatype	NULL	Description
MAXQUERYLEN	NUMBER		Identifies the length of the longest query (in number of seconds) executed in the instance during the period. You can use this statistic to estimate the proper setting of the UNDO_RETENTION initialization parameter. The length of a query is measured from the cursor open time to the last fetch/execute time of the cursor. Only the length of those cursors that have been fetched/executed during the period are reflected in the view.
MAXQUERYSQLID	VARCHAR2(13)		SQL identifier of the longest running SQL statement in the period
MAXCONCURRENCY	NUMBER		Identifies the highest number of transactions executed concurrently within the period
UNXPSTEALCNT	NUMBER		Number of attempts to obtain undo space by stealing unexpired extents from other transactions
UNXPBLKRELCNT	NUMBER		Number of unexpired blocks removed from certain undo segments so they can be used by other transactions
UNXPBLKREUCNT	NUMBER		Number of unexpired undo blocks reused by transactions
EXPSTEALCNT	NUMBER		Number of attempts to steal expired undo blocks from other undo segments
EXPBLKRELCNT	NUMBER		Number of expired undo blocks stolen from other undo segments
EXPBLKREUCNT	NUMBER		Number of expired undo blocks reused within the same undo segments
SSOLDERRCNT	NUMBER		Identifies the number of times the error ORA-01555 occurred. You can use this statistic to decide whether or not the UNDO_RETENTION initialization parameter is set properly given the size of the undo tablespace. Increasing the value of UNDO_RETENTION can reduce the occurrence of this error.
NOSPACEERRCNT	NUMBER		Identifies the number of times space was requested in the undo tablespace and there was no free space available. That is, all of the space in the undo tablespace was in use by active transactions. The corrective action is to add more space to the undo tablespace.
ACTIVEBLKS	NUMBER		Total number of blocks in the active extents of the undo tablespace for the instance at the sampled time in the period
UNEXPIREDBLKS	NUMBER		Total number of blocks in the unexpired extents of the undo tablespace for the instance at the sampled time in the period
EXPIREDBLKS	NUMBER		Total number of blocks in the expired extents of the undo tablespace for the instance at the sampled time in the period
TUNED_UNDORETENTION	NUMBER		System tuned value indicating the period for which undo is being retained

DBA_HIST_WAITCLASSMET_HISTORY

DBA_HIST_WAITCLASSMET_HISTORY displays the history of the wait event class metric data kept by the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot

Column	Datatype	NULL	Description
WAIT_CLASS_ID	NUMBER	NOT NULL	Identifier for the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name for the class of the wait event
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Metric group ID
AVERAGE_WAITER_COUNT	NUMBER	NOT NULL	Average waiter count
DBTIME_IN_WAIT	NUMBER	NOT NULL	Database time spent in the wait
TIME_WAITED	NUMBER	NOT NULL	Time waited during this interval
WAIT_COUNT	NUMBER	NOT NULL	Number of times waited

DBA_HIST_WAITSTAT

DBA_HIST_WAITSTAT displays historical block contention statistics. This view contains snapshots of V\$WAITSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
CLASS	VARCHAR2(18)		Class of the block
WAIT_COUNT	NUMBER		Number of waits by the OPERATION for this CLASS of block
TIME	NUMBER		Sum of all wait times for all the waits by the OPERATION for this CLASS of block

DBA_HIST_WR_CONTROL

DBA_HIST_WR_CONTROL displays the control information for the Workload Repository.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SNAP_INTERVAL	INTERVAL DAY(5) TO SECOND(1)	NOT NULL	Snapshot interval; how often to automatically take snapshots
RETENTION	INTERVAL DAY(5) TO SECOND(1)	NOT NULL	Retention setting for the snapshots; amount of time to keep the snapshots

DBA_IND_COLUMNS

DBA_IND_COLUMNS describes the columns of all the indexes on all tables and clusters in the database. Its columns are the same as those in ["ALL_IND_COLUMNS"](#) on page 2-35.

DBA_IND_EXPRESSIONS

DBA_IND_EXPRESSIONS lists expressions of function-based indexes on all tables and clusters in the database. Its columns are the same as those in ["ALL_IND_EXPRESSIONS"](#) on page 2-36.

DBA_IND_PARTITIONS

DBA_IND_PARTITIONS describes, for each index partition in the database, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in ["ALL_IND_PARTITIONS"](#) on page 2-36.

DBA_IND_STATISTICS

DBA_IND_STATISTICS displays optimizer statistics for all indexes in the database. Its columns are the same as those in ALL_IND_STATISTICS.

See Also: ["ALL_IND_STATISTICS"](#) on page 2-37

DBA_IND_SUBPARTITIONS

DBA_IND_SUBPARTITIONS describes, for each index subpartition in the database, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in ["ALL_IND_SUBPARTITIONS"](#) on page 2-39.

DBA_INDEXES

DBA_INDEXES describes all indexes in the database. To gather statistics for this view, use the SQL ANALYZE statement. This view supports parallel partitioned index scans. Its columns are the same as those in ["ALL_INDEXES"](#) on page 2-40.

DBA_INDEXTYPE_ARRAYTYPES

DBA_INDEXTYPE_ARRAYTYPES displays information about the array types specified by all indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_ARRAYTYPES.

See Also: ["ALL_INDEXTYPE_ARRAYTYPES"](#) on page 2-43

DBA_INDEXTYPE_COMMENTS

DBA_INDEXTYPE_COMMENTS displays comments for all user-defined indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_COMMENTS.

See Also: ["ALL_INDEXTYPE_COMMENTS"](#) on page 2-43

DBA_INDEXTYPE_OPERATORS

DBA_INDEXTYPE_OPERATORS lists all the operators supported by indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_OPERATORS.

See Also: ["ALL_INDEXTYPE_OPERATORS"](#) on page 2-44

DBA_INDEXTYPES

DBA_INDEXTYPES describes all indextypes in the database. Its columns are the same as those in ALL_INDEXTYPES.

See Also: ["ALL_INDEXTYPES"](#) on page 2-44

DBA_INTERNAL_TRIGGERS

DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database. Its columns are the same as those in ["ALL_INTERNAL_TRIGGERS"](#) on page 2-45.

DBA_JAVA_ARGUMENTS

DBA_JAVA_ARGUMENTS displays argument information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_ARGUMENTS.

See Also: ["ALL_JAVA_ARGUMENTS"](#) on page 2-45

DBA_JAVA_CLASSES

DBA_JAVA_CLASSES displays class level information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_CLASSES.

See Also: ["ALL_JAVA_CLASSES"](#) on page 2-46

DBA_JAVA_DERIVATIONS

DBA_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for all Java classes in the database. Its columns are the same as those in ALL_JAVA_DERIVATIONS.

See Also: ["ALL_JAVA_DERIVATIONS"](#) on page 2-47

DBA_JAVA_FIELDS

DBA_JAVA_FIELDS displays field information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_FIELDS.

See Also: ["ALL_JAVA_FIELDS"](#) on page 2-47

DBA_JAVA_IMPLEMENTATIONS

DBA_JAVA_IMPLEMENTATIONS describes interfaces implemented by all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_IMPLEMENTATIONS.

See Also: ["ALL_JAVA_IMPLEMENTATIONS"](#) on page 2-48

DBA_JAVA_INNERS

DBA_JAVA_INNERS displays information about inner classes referred to by all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_INNERS.

See Also: ["ALL_JAVA_INNERS"](#) on page 2-49

DBA_JAVA_LAYOUTS

DBA_JAVA_LAYOUTS displays class layout information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_LAYOUTS.

See Also: ["ALL_JAVA_LAYOUTS"](#) on page 2-49

DBA_JAVA_METHODS

DBA_JAVA_METHODS displays method information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_METHODS.

See Also: ["ALL_JAVA_METHODS"](#) on page 2-50

DBA_JAVA_NCOMPS

DBA_JAVA_NCOMPS displays ncomp-related information about all Java classes in the database. Its columns are the same as those in ALL_JAVA_NCOMPS.

See Also: ["ALL_JAVA_NCOMPS"](#) on page 2-51

DBA_JAVA_POLICY

DBA_JAVA_POLICY describes Java security permissions for all users in the database.

Related View

USER_JAVA_POLICY describes Java security permissions for the current user.

Column	Datatype	NULL	Description
KIND	VARCHAR2(8)		Indicates whether the permission is a positive (GRANT) or a limitation (RESTRICT)
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user, schema, or role to which the permission object is assigned
TYPE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema in which the permission object is loaded
TYPE_NAME	VARCHAR2(4000)		Permission class type, which is designated by a string containing the full class name, such as, <code>java.io.FilePermission</code>
NAME	VARCHAR2(4000)		Target attribute (name) of the permission object. This name is used when defining the permission.
ACTION	VARCHAR2(4000)		Action attribute for this permission. Many permissions expect a null value if no action is appropriate for the permission.
ENABLED	VARCHAR2(8)		Indicates whether the permission is enabled (ENABLED) or disabled (DISABLED)
SEQ	NUMBER		Sequence number used to identify this row. This number should be supplied when disabling, enabling, or deleting the permission.

See Also: ["USER_JAVA_POLICY"](#) on page 3-168

DBA_JAVA_RESOLVERS

DBA_JAVA_RESOLVERS displays information about resolvers of all Java classes in the database. Its columns are the same as those in ALL_JAVA_RESOLVERS.

See Also: ["ALL_JAVA_RESOLVERS"](#) on page 2-52

DBA_JAVA_THROWS

DBA_JAVA_THROWS displays information about exceptions thrown from methods of all Java classes in the database. Its columns are the same as those in ALL_JAVA_THROWSa.

See Also: ["ALL_JAVA_THROWS"](#) on page 2-52

DBA_JOBS

DBA_JOBS describes all jobs in the database. Its columns are the same as those in ["ALL_JOBS"](#) on page 2-53.

See Also: *Oracle Database Administrator's Guide* for more information on jobs

DBA_JOBS_RUNNING

DBA_JOBS_RUNNING lists all jobs in the database that are currently running.

Column	Datatype	NULL	Description
SID	NUMBER		Identifier of process that is executing the job. See "V\$LOCK" on page 4-76.
JOB	NUMBER		Identifier of job. This job is currently executing.
FAILURES	NUMBER		Number of times this job started and failed since its last success.
LAST_DATE	DATE		Date that this job last successfully executed.
LAST_SEC	VARCHAR2(8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing.
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE. This is when the last successful execution started.
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0.

DBA_JOIN_IND_COLUMNS

DBA_JOIN_IND_COLUMNS describes all join conditions in the database. Its columns are the same as those in ["ALL_JOIN_IND_COLUMNS"](#) on page 2-54.

DBA_KGLLOCK

DBA_KGLLOCK lists all the locks and pins held on KGL objects (objects in the Kernel Generic Library cache).

Column	Datatype	NULL	Description
kgllkuse	RAW(4)		Address of the user session that holds the lock or pin
kgllkhdl	RAW(4)		Address of the handle for the KGL object
kgllkmod	NUMBER		Current mode of the lock or pin
kgllkreq	NUMBER		Mode in which the lock or pin was requested
kgllktype	VARCHAR2(4)		Whether this is a lock or a pin

DBA_LIBRARIES

DBA_LIBRARIES describes all the libraries in the database. Its columns are the same as those in ["ALL_LIBRARIES"](#) on page 2-54.

DBA_LMT_FREE_SPACE

DBA_LMT_FREE_SPACE describes the free extents in all locally managed tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_ID	NUMBER		Identifier number of the tablespace containing the extent
FILE_ID	NUMBER		File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Starting block number of the extent
BLOCKS	NUMBER		Size of the extent (in Oracle blocks)

DBA_LMT_USED_EXTENTS

DBA_LMT_USED_EXTENTS describes the extents comprising the segments in all locally managed tablespaces in the database.

Column	Datatype	NULL	Description
SEGMENT_FILEID	NUMBER		File number of the segment header of the extent
SEGMENT_BLOCK	NUMBER		Block number of the segment header of the extent
TABLESPACE_ID	NUMBER		Identifier number of the tablespace containing the extent
EXTENT_ID	NUMBER		Extent number in the segment
FILEID	NUMBER		File identifier number of the file containing the extent
BLOCK	NUMBER		Starting block number of the extent
LENGTH	NUMBER		Number of blocks in the extent

DBA_LOB_PARTITIONS

DBA_LOB_PARTITIONS displays all LOB partitions in the database. Its columns are the same as those in ["ALL_LOB_PARTITIONS"](#) on page 2-54.

DBA_LOB_SUBPARTITIONS

DBA_LOB_SUBPARTITIONS displays partition-level attributes of all LOB data subpartitions in the database. Its columns are the same as those in ["ALL_LOB_SUBPARTITIONS"](#) on page 2-55.

DBA_LOB_TEMPLATES

DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database. Its columns are the same as those in [ALL_LOB_TEMPLATES](#).

See Also: ["ALL_LOB_TEMPLATES"](#) on page 2-56

DBA_LOBS

DBA_LOBS displays the BLOBs and CLOBs contained in all tables in the database. BFILEs are stored outside the database, so they are not described by this view. This view's columns are the same as those in ["ALL_LOBS"](#) on page 2-57.

DBA_LOCK

DBA_LOCK lists all locks or latches held in the database, and all outstanding requests for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(26)		Lock type See Also: For a listing of lock types, see Appendix D, "Oracle Enqueue Names"
MODE_HELD	VARCHAR2(40)		Lock mode
MODE_REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(40)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Whether the lock is currently blocking others

DBA_LOCK_INTERNAL

DBA_LOCK_INTERNAL displays a row for each lock or latch that is being held, and one row for each outstanding request for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(56)		Lock type See Also: For a listing of lock types, see Appendix D, "Oracle Enqueue Names"
MODE_HELD	VARCHAR2(40)		Lock mode
MODE_REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(1130)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2

DBA_LOCKS

DBA_LOCKS is a synonym for DBA_LOCK.

See Also: ["DBA_LOCK"](#) on page 3-87

DBA_LOG_GROUP_COLUMNS

DBA_LOG_GROUP_COLUMNS describes all columns in the database that are specified in log groups. Its columns are the same as those in ["ALL_LOG_GROUP_COLUMNS"](#) on page 2-58.

DBA_LOG_GROUPS

DBA_LOG_GROUPS describes log group definitions on all tables in the database. Its columns are the same as those in ["ALL_LOG_GROUPS"](#) on page 2-58.

DBA_LOGMNR_LOG

DBA_LOGMNR_LOG displays all archived logs registered with active LogMiner persistent sessions in the database.

A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.

Column	Datatype	NULL	Description
LOGMNR_SESSION_ID	NUMBER	NOT NULL	Unique identifier of the persistent session
NAME	VARCHAR2(513)		Name of the archived log
DBID	NUMBER	NOT NULL	Database identifier that produced the archived log
RESETLOGS_SCN	NUMBER	NOT NULL	SCN at which resetlogs operation was performed at the source database generating the archived log
RESETLOGS_TIME	NUMBER	NOT NULL	Timestamp at which resetlogs operation was performed at the source database generating the archived log
MODIFIED_TIME	DATE		Time at which the archived log was registered with LogMiner
THREAD#	NUMBER	NOT NULL	Redo thread at the source database that generated the archived log
SEQUENCE#	NUMBER	NOT NULL	Logfile sequence number
FIRST_SCN	NUMBER	NOT NULL	Lowest SCN of the redo record contained in the logfile
NEXT_SCN	NUMBER	NOT NULL	Highest possible SCN of the redo record contained in the logfile
FIRST_TIME	DATE		Time of the first redo record contained in the logfile
NEXT_TIME	DATE		Time of the last redo record contained in the logfile
DICTIONARY_BEGIN	VARCHAR2(3)		Indicates whether the archived log contains the beginning of a LogMiner dictionary (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)		Indicates whether the archived log contains the end of a LogMiner dictionary (YES) or not (NO)
KEEP	VARCHAR2(3)		Indicates whether the logfile is still required for this LogMiner session (YES) or not (NO)
SUSPECT	VARCHAR2(3)		Indicates whether the archived log content was deemed to be corrupt or the archived log is partially filled (YES) or not (NO)

DBA_LOGMNR_PURGED_LOG

DBA_LOGMNR_PURGED_LOG displays archived logs that are no longer needed by LogMiner persistent sessions active at the database.

A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Log file name

DBA_LOGMNR_SESSION

DBA_LOGMNR_SESSION displays all active LogMiner persistent sessions in the database.

A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Unique session identifier
NAME	VARCHAR2(128)	NOT NULL	Unique session name
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database whose archived logs are to be mined in this persistent LogMiner session
SOURCE_DBID	NUMBER		Database ID of the source database
SOURCE_RESETLOGS_SCN	NUMBER		Resetlogs SCN associated with the incarnation of the source database whose archived logs are mined
SOURCE_RESETLOGS_TIME	NUMBER		Resetlogs time associated with the incarnation of the source database whose archived logs are mined
FIRST_SCN	NUMBER		Only modifications that occurred on or after this SCN can be mined using this persistent session
END_SCN	NUMBER		No modifications that occurred on or after this SCN can be mined using this persistent session
BRANCH_SCN	NUMBER		SCN at which a branch will be taken in terms of the incarnation corresponding to the source database. This implies a point-in-time recovery was performed at the source database at this SCN.
WAIT_FOR_LOG	VARCHAR2(3)		Indicates whether the persistent session waits for RFS to register new archived logs or to fill gaps (YES) or not (NO)
HOT_MINE	VARCHAR2(3)		Indicates whether real-time mining is on (YES) or not (NO)
SAFE_PURGE_SCN	NUMBER		Persistent session can safely be purged up to this SCN
CHECKPOINT_SCN	NUMBER		SCN at which the latest checkpoint is taken by the persistent logminer session

DBA_LOGSTDBY_EVENTS

DBA_LOGSTDBY_EVENTS displays information about the activity of the logical standby database system. It can be used to determine the cause of failures that occur when applying redo data to logical standby databases. This view is for logical standby databases only.

Column	Datatype	NULL	Description
EVENT_TIME	DATE		Time the event was logged
CURRENT_SCN	NUMBER		Change vector SCN for the change. If a failure occurred, then examine this column to determine which archived log file contains the source of the failure (for example, an unsupported record).
COMMIT_SCN	NUMBER		SCN value on which the change was committed
XIDUSN	NUMBER		Transaction ID undo segment number
XIDSLT	NUMBER		Transaction ID slot number
XIDSQN	NUMBER		Transaction ID sequence number
EVENT	CLOB		Statement that was being processed when the failure occurred
STATUS_CODE	NUMBER		Status (or Oracle error code) belonging to the STATUS message
STATUS	VARCHAR2(2000)		Description of the current activity of the process or the reason why the apply operation stopped

DBA_LOGSTDBY_LOG

DBA_LOGSTDBY_LOG displays information about the logs registered for a logical standby database. This view is for logical standby databases only.

Column	Datatype	NULL	Description
THREAD#	NUMBER	NOT NULL	Thread ID of the archive log. The THREAD number is 1 for a single instance. For Real Application Clusters, this column will contain different numbers.
SEQUENCE#	NUMBER	NOT NULL	Sequence number of the archive log file
FIRST_CHANGE#	NUMBER	NOT NULL	System change number (SCN) of the current archive log
NEXT_CHANGE#	NUMBER	NOT NULL	SCN of the next archive log
FIRST_TIME	DATE		Date of the current archive log
NEXT_TIME	DATE		Date of the next archive log
FILE_NAME	VARCHAR2(513)		Name of the archive log
TIMESTAMP	DATE		Time when the archive log was registered
DICTION_BEGIN	VARCHAR2(3)		Indicates whether the beginning of the dictionary build is in this archive log (YES) or not (NO)
DICTION_END	VARCHAR2(3)		Indicates whether the end of the dictionary build is in this archive log (YES) or not (NO)
APPLIED	VARCHAR2(7)		Indicates whether each archive log has been applied (YES) or not (NO)

Note: The SCN values in this view correlate to the SCN values shown in the DBA_LOGSTDBY_PROGRESS view.

DBA_LOGSTDBY_NOT_UNIQUE

DBA_LOGSTDBY_NOT_UNIQUE displays all tables that have no primary and no non-null unique indexes. Most of the tables displayed by this view are supported because their columns contain enough information to be maintained in a logical standby database. Some tables, however, cannot be supported because their columns do not contain the necessary information. Unsupported tables usually contain a column defined using an unsupported datatype.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema name of the non-unique table
TABLE_NAME	VARCHAR2(30)		Table name of the non-unique table
BAD_COLUMN	VARCHAR2(1)		<ul style="list-style-type: none"> ■ Y - Table column is defined using an unbounded data type, such as LONG or BLOB. If two rows in the table match except in their LOB columns, then the table cannot be maintained properly. Log apply services will attempt to maintain these tables, but you must ensure the application does not allow uniqueness only in the unbounded columns. ■ N - Enough column information is present to maintain the table in the logical standby database but the log transport services and log apply services would run more efficiently if you added a primary key. You should consider adding a disabled RELY constraint to these tables.

DBA_LOGSTDBY_PARAMETERS

DBA_LOGSTDBY_PARAMETERS displays the list of parameters used by SQL apply for logical standby databases. This view is for logical standby databases only.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the parameter: <ul style="list-style-type: none"> ▪ MAX_SGA - System global area (SGA) allocated for the log apply services cache (in megabytes) ▪ MAX_SLAVES - Number of parallel query servers specifically reserved for log apply services ▪ MAX_EVENTS_RECORDED - Number of events stored in the DBA_LOGSTDBY_EVENTS view ▪ TRANSACTION_CONSISTENCY - Shows the level of transaction consistency maintained (FULL, READ_ONLY, or NONE) ▪ RECORD_SKIP_ERRORS - Indicates records that are skipped ▪ RECORD_SKIP_DDL - Indicates skipped DDL statements ▪ RECORD_APPLIED_DDL - Indicates applied DDL statements ▪ FIRST_SCN - SCN at which log transport services will begin reading redo information ▪ PRIMARY - Database ID of the database to which logs are being applied ▪ LMNR_SID - LogMiner Session ID. This internal value indicates which LogMiner session is in use. ▪ UNTIL_SCN - SCN value at which the log apply services will shut down all transactions that have been applied ▪ END_PRIMARY_SCN - During a switchover, this value indicates the last SCN applied by the new primary from the old primary ▪ NEW_PRIMARY_SCN - During a switchover, this value indicates the starting SCN for the new primary database ▪ COMPLETED_SESSION - Indicates that the log apply services session has concluded. The value will indicate SWITCHOVER or FAILOVER, as appropriate. ▪ PREPARE_SERVERS - Controls the number of parallel execution servers used to prepare changes ▪ APPLY_SERVERS - Controls the number of parallel execution servers used to apply changes ▪ APPLY_SCN - SCN value after which apply servers will begin applying redo information
VALUE	VARCHAR2(2000)		Value of the parameter

DBA_LOGSTDBY_PROGRESS

DBA_LOGSTDBY_PROGRESS displays the progress of log apply services on the logical standby database. This view is for logical standby databases only.

Column	Datatype	NULL	Description
APPLIED_SCN	NUMBER		Newest SCN at which all changes have been applied. The values in the APPLIED_SCN and NEWEST_SCN columns will match if all available redo log data has been processed.

Column	Datatype	NULL	Description
APPLIED_THREAD#	NUMBER		Thread number for a log containing the APPLIED_SCN
APPLIED_SEQUENCE#	NUMBER		Sequence number for a log containing the APPLIED_SCN
APPLIED_TIME	DATE		Estimate of the time and date of the APPLIED_SCN
READ_SCN	NUMBER		All log data less than this SCN has been read and saved
READ_THREAD#	NUMBER		Thread number for a log containing the READ_SCN
READ_SEQUENCE#	NUMBER		Sequence number for a log containing the READ_SCN
READ_TIME	DATE		Estimate of the time and date of the READ_SCN
NEWEST_SCN	NUMBER		Most recent SCN available on the standby system. If no more logs are being shipped, then changes could be applied to this SCN. The values in the APPLIED_SCN and NEWEST_SCN columns will match if all available redo log data has been processed.
NEWEST_THREAD#	NUMBER		Thread number for a log containing the NEWEST_SCN
NEWEST_SEQUENCE#	NUMBER		Sequence number for a log containing the NEWEST_SCN
NEWEST_TIME	DATE		Estimate of the time and date of the NEWEST_SCN

Note: The SCN values shown in this view correlate to the SCN values shown in the DBA_LOGSTDBY_LOG view.

DBA_LOGSTDBY_SKIP

DBA_LOGSTDBY_SKIP displays the tables that will be skipped by log apply services. This view is for logical standby databases only.

Column	Datatype	NULL	Description
ERROR	VARCHAR2(1)		Indicates whether the statement should be skipped (Y) or just return errors for the statement (N)
STATEMENT_OPT	VARCHAR2(30)		Specifies the type of statement that should be skipped. It must be one of the SYSTEM_AUDIT statement options.
OWNER	VARCHAR2(30)		Name of the schema under which the skip option should be used
NAME	VARCHAR2(30)		Name of the option under which the skip option should be used
USE_LIKE	VARCHAR2(1)		Indicates whether the statement should use a SQL wildcard search when matching names (Y) or not (N)
ESC	VARCHAR2(1)		Escape character used when performing wildcard matches
PROC	VARCHAR2(98)		Name of a stored procedure that will be executed when processing the skip option

DBA_LOGSTDBY_SKIP_TRANSACTION

DBA_LOGSTDBY_SKIP_TRANSACTION displays the skip settings chosen. This view is for logical standby databases only.

Column	Datatype	NULL	Description
XIDUSN	NUMBER		Transaction ID undo segment number
XIDSLT	NUMBER		Transaction ID slot number

Column	Datatype	NULL	Description
XIDSQN	NUMBER		Transaction ID sequence number

DBA_LOGSTDBY_UNSUPPORTED

DBA_LOGSTDBY_UNSUPPORTED displays the schemas, tables, and columns in those tables that contain unsupported datatypes. Use this view when you are preparing to create a logical standby database. This view is for logical standby databases only.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Schema name of the unsupported table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the unsupported table
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the unsupported column
DATA_TYPE	VARCHAR2(106)		Datatype of the unsupported column
ATTRIBUTES	VARCHAR2(39)		Indicates unsupported storage attributes for a given table: <ul style="list-style-type: none"> ▪ IOT with Overflow ▪ IOT with LOB ▪ IOT with physical Rowid mapping ▪ Mapping table for physical rowid of IOT ▪ IOT with row movement ▪ Segment Compression

DBA_METHOD_PARAMS

DBA_METHOD_PARAMS describes the method parameters of all object types in the database. Its columns are the same as those in ALL_METHOD_PARAMS.

See Also: ["ALL_METHOD_PARAMS"](#) on page 2-59

DBA_METHOD_RESULTS

DBA_METHOD_RESULTS describes the method results of all object types in the database. Its columns are the same as those in ALL_METHOD_RESULTS.

See Also: ["ALL_METHOD_RESULTS"](#) on page 2-59

DBA_MVIEW_AGGREGATES

DBA_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of all aggregated materialized view in the database. Its columns are the same as those in ["ALL_MVIEW_AGGREGATES"](#) on page 2-60.

DBA_MVIEW_ANALYSIS

DBA_MVIEW_ANALYSIS describes all materialized views in the database that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in ["ALL_MVIEW_ANALYSIS"](#) on page 2-61.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as `SYSDATE` or `USER`. This view also excludes materialized views that were created as snapshots prior to Oracle8i and that were never altered to enable query rewrite.

DBA_MVIEW_COMMENTS

`DBA_MVIEW_COMMENTS` displays comments on all materialized views in the database. Its columns are the same as those in `ALL_MVIEW_COMMENTS`.

See Also: ["ALL_MVIEW_COMMENTS"](#) on page 2-62

DBA_MVIEW_DETAIL_RELATIONS

`DBA_MVIEW_DETAIL_RELATIONS` represents the named detail relations that are either in the `FROM` list of a materialized view, or that are indirectly referenced through views in the `FROM` list. Its columns are the same as those in ["ALL_MVIEW_DETAIL_RELATIONS"](#) on page 2-62.

DBA_MVIEW_JOINS

`DBA_MVIEW_JOINS` describes a join between two columns in the `WHERE` clause of a subquery that defines a materialized view. Its columns are the same as those in ["ALL_MVIEW_JOINS"](#) on page 2-63.

DBA_MVIEW_KEYS

`DBA_MVIEW_KEYS` describes the columns or expressions in the `SELECT` list upon which materialized views in the database are based. Its columns are the same as those in ["ALL_MVIEW_KEYS"](#) on page 2-63.

DBA_MVIEW_LOG_FILTER_COLS

`DBA_MVIEW_LOG_FILTER_COLS` lists all columns (excluding primary key columns) being logged in the materialized view logs.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the master table being logged
NAME	VARCHAR2(30)	NOT NULL	Name of the master table being logged
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column being logged

DBA_MVIEW_LOGS

`DBA_MVIEW_LOGS` describes all materialized view logs in the database. Its columns are the same as those in `ALL_MVIEW_LOGS`.

See Also: ["ALL_MVIEW_LOGS"](#) on page 2-64

DBA_MVIEW_REFRESH_TIMES

DBA_MVIEW_REFRESH_TIMES describes refresh times of all materialized views in the database. Its columns are the same as those in ALL_MVIEW_REFRESH_TIMES.

See Also: ["ALL_MVIEW_REFRESH_TIMES"](#) on page 2-65

DBA_MVIEWS

DBA_MVIEWS describes all materialized views in the database. Its columns are the same as those in ALL_MVIEWS.

See Also: ["ALL_MVIEWS"](#) on page 2-65

DBA_NESTED_TABLE_COLS

DBA_NESTED_TABLE_COLS describes the columns of all nested tables in the database. Its columns are the same as those in ["ALL_NESTED_TABLE_COLS"](#) on page 2-68. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_NESTED_TABLES

DBA_NESTED_TABLES describes all nested tables contained in all tables in the database. Its columns are the same as those in ["ALL_NESTED_TABLES"](#) on page 2-70.

DBA_OBJ_AUDIT_OPTS

DBA_OBJ_AUDIT_OPTS describes auditing options on all objects.

Related View

USER_OBJ_AUDIT_OPTS describes auditing options on all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the object
OBJECT_NAME	VARCHAR2 (30)		Name of the object
OBJECT_TYPE	VARCHAR2 (9)		Type of the object
ALT	VARCHAR2 (3)		Auditing ALTER WHENEVER SUCCESSFUL/UNSUCCESSFUL
AUD	VARCHAR2 (3)		Auditing AUDIT WHENEVER SUCCESSFUL/UNSUCCESSFUL
COM	VARCHAR2 (3)		Auditing COMMENT WHENEVER SUCCESSFUL/UNSUCCESSFUL
DEL	VARCHAR2 (3)		Auditing DELETE WHENEVER SUCCESSFUL/UNSUCCESSFUL
GRA	VARCHAR2 (3)		Auditing GRANT WHENEVER SUCCESSFUL/UNSUCCESSFUL
IND	VARCHAR2 (3)		Auditing INDEX WHENEVER SUCCESSFUL/UNSUCCESSFUL
INS	VARCHAR2 (3)		Auditing INSERT WHENEVER SUCCESSFUL/UNSUCCESSFUL
LOC	VARCHAR2 (3)		Auditing LOCK WHENEVER SUCCESSFUL/UNSUCCESSFUL

Column	Datatype	NULL	Description
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL/UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL/UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL/UNSUCCESSFUL
REF	CHAR(3)		Auditing REFERENCE WHENEVER SUCCESSFUL/UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL/UNSUCCESSFUL
CRE	VARCHAR2(3)		Auditing CREATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing READ WHENEVER SUCCESSFUL/UNSUCCESSFUL
WRI	VARCHAR2(3)		Auditing WRITE WHENEVER SUCCESSFUL / UNSUCCESSFUL
FBK	VARCHAR2(3)		Auditing FLASHBACK WHENEVER SUCCESSFUL / UNSUCCESSFUL

DBA_OBJ_COLATTRS

DBA_OBJ_COLATTRS describes object columns and attributes contained in all tables in the database. Its columns are the same as those in ALL_OBJ_COLATTRS.

See Also: ["ALL_OBJ_COLATTRS"](#) on page 2-70

DBA_OBJECT_SIZE

DBA_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects.

Related View

- USER_OBJECT_SIZE lists the size of PL/SQL objects owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: TYPE, TYPE BODY, TABLE, VIEW, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, JAVA SOURCE, JAVA CLASS or JAVA RESOURCE
SOURCE_SIZE	NUMBER		Size of the source in bytes. Must be in memory during compilation, or dynamic recompilation.
PARSED_SIZE	NUMBER		Size of the parsed form of the object, in bytes. Must be in memory when an object is being compiled that references this object.
CODE_SIZE	NUMBER		Code size, in bytes. Must be in memory when this object is executing.
ERROR_SIZE	NUMBER		Size of error messages, in bytes. In memory during the compilation of the object when there are compilation errors.

DBA_OBJECT_TABLES

DBA_OBJECT_TABLES describes all object tables in the database. Its columns are the same as those in ALL_OBJECT_TABLES.

See Also: ["ALL_OBJECT_TABLES"](#) on page 2-71

DBA_OBJECTS

DBA_OBJECTS describes all objects in the database. Its columns are the same as those in ["ALL_OBJECTS"](#) on page 2-73.

DBA_OPANCILLARY

DBA_OPANCILLARY provides ancillary information for all operator bindings in the database. Its columns are the same as those in ["ALL_OPANCILLARY"](#) on page 2-74.

DBA_OPARGUMENTS

DBA_OPARGUMENTS provides argument information for all operator bindings in the database. Its columns are the same as those in ["ALL_OPARGUMENTS"](#) on page 2-74.

DBA_OPBINDINGS

DBA_OPBINDINGS describes the binding functions and methods on all operators in the database. Its columns are the same as those in ALL_OPBINDINGS.

See Also: ["ALL_OPBINDINGS"](#) on page 2-75

DBA_OPERATOR_COMMENTS

DBA_OPERATOR_COMMENTS displays comments for all user-defined operators in the database. Its columns are the same as those in ALL_OPERATOR_COMMENTS.

See Also: ["ALL_OPERATOR_COMMENTS"](#) on page 2-75

DBA_OPERATORS

DBA_OPERATORS describes all operators in the database. Its columns are the same as those in ALL_OPERATORS.

See Also: ["ALL_OPERATORS"](#) on page 2-76

DBA_ORPHAN_KEY_TABLE

DBA_ORPHAN_KEY_TABLE reports key values from indexes where the underlying base table has block corruptions. To create the view, run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the orphan key table for an index, run the DBMS_REPAIR.DUMP_ORPHAN_KEYS procedure on the index. For each key in the index that points to a corrupt data block, Oracle inserts a row into the orphan key table.

Column	Datatype	NULL	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index

Column	Datatype	NULL	Description
IPART_NAME	VARCHAR2(30)	NULL	Name of the index partition or subpartition
INDEX_ID	NUMBER	NOT NULL	Dictionary object ID of the index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the base table of the index
PART_NAME	VARCHAR2(30)	NULL	Name of the base table partition or subpartition
TABLE_ID	NUMBER	NOT NULL	Dictionary object ID of the base table
KEYDATA	UROWID	NOT NULL	Physical rowid of the corrupt data row
KEY	UROWID	NOT NULL	Key values for the index entry
DUMP_TIMESTAMP	DATE	NOT NULL	Timestamp when the entry was made into the orphan key table

DBA_OUTLINE_HINTS

DBA_OUTLINE_HINTS describes the set of hints that make up all the outlines in the database. Its columns are the same as those in ["ALL_OUTLINE_HINTS"](#) on page 2-76.

DBA_OUTLINES

DBA_OUTLINES describes all outlines in the database. Its columns are the same as those in ["ALL_OUTLINES"](#) on page 2-77.

DBA_OUTSTANDING_ALERTS

DBA_OUTSTANDING_ALERTS describes alerts which the server considers to be outstanding.

Column	Datatype	NULL	Description
SEQUENCE_ID	NUMBER		Alert sequence number
REASON_ID	NUMBER	NOT NULL	ID of the alert reason
OWNER	VARCHAR2(30)		Owner of the object on which the alert was issued
OBJECT_NAME	VARCHAR2(513)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(64)		Object type
REASON	VARCHAR2(4000)		Reason for the alert
TIME_SUGGESTED	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was last updated
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was first created
SUGGESTED_ACTION	VARCHAR2(4000)		Advice of the recommended action
ADVISOR_NAME	VARCHAR2(30)		Name of the advisor to be invoked for more information
METRIC_VALUE	NUMBER		Value of the related metrics
MESSAGE_TYPE	VARCHAR2(12)		Message type: <ul style="list-style-type: none"> ■ Notification ■ Warning
MESSAGE_GROUP	VARCHAR2(64)		Name of the message group to which the alert belongs
MESSAGE_LEVEL	NUMBER		Message severity level (1 to 32)
HOSTING_CLIENT_ID	VARCHAR2(64)		ID of the client or security group to which the alert relates

Column	Datatype	NULL	Description
MODULE_ID	VARCHAR2(64)		ID of the module that originated the alert
PROCESS_ID	VARCHAR2(128)		Process ID
HOST_ID	VARCHAR2(256)		DNS host name of the originating host
HOST_NW_ADDR	VARCHAR2(256)		IP or other network address of the originating host
INSTANCE_NAME	VARCHAR2(15)		Originating instance name
INSTANCE_NUMBER	NUMBER		Originating instance number
USER_ID	VARCHAR2(30)		User ID
EXECUTION_CONTEXT_ID	VARCHAR2(60)		ID of the threshold of execution
ERROR_INSTANCE_ID	VARCHAR2(142)		ID of an error instance plus a sequence number

DBA_PART_COL_STATISTICS

DBA_PART_COL_STATISTICS provides column statistics and histogram information for partitions of all tables in the database. Its columns are the same as those in "ALL_PART_COL_STATISTICS" on page 2-77.

DBA_PART_HISTOGRAMS

DBA_PART_HISTOGRAMS provides the histogram data (end-points per histogram) for histograms on all table partitions in the database. Its columns are the same as those in "ALL_PART_HISTOGRAMS" on page 2-78.

DBA_PART_INDEXES

DBA_PART_INDEXES displays the object-level partitioning information for all partitioned indexes in the database. Its columns are the same as those in ALL_PART_INDEXES.

See Also: "ALL_PART_INDEXES" on page 2-79

DBA_PART_KEY_COLUMNS

DBA_PART_KEY_COLUMNS describes the partitioning key columns for all partitioned objects in the database. Its columns are the same as those in ALL_PART_KEY_COLUMNS.

See Also: "ALL_PART_KEY_COLUMNS" on page 2-80

DBA_PART_LOBS

DBA_PART_LOBS provides table-level information for all partitioned LOBs in the database, including default attributes for LOB data partitions. Its columns are the same as those in "ALL_PART_LOBS" on page 2-81.

DBA_PART_TABLES

DBA_PART_TABLES displays the object-level partitioning information for all partitioned tables in the database. Its columns are the same as those in ALL_PART_TABLES.

See Also: ["ALL_PART_TABLES"](#) on page 2-82

DBA_PARTIAL_DROP_TABS

DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations. Its columns are the same as those in ["ALL_PARTIAL_DROP_TABS"](#) on page 2-83.

DBA_PENDING_CONV_TABLES

DBA_PENDING_CONV_TABLES describes all pending conversion tables in the database. Its columns are the same as those in ALL_PENDING_CONV_TABLES.

See Also: ["ALL_PENDING_CONV_TABLES"](#) on page 2-83

DBA_PENDING_TRANSACTIONS

DBA_PENDING_TRANSACTIONS describes unresolved transactions (either due to failure or if the coordinator has not sent a commit/rollback).

Column	Datatype	NULL	Description
FORMATID	NUMBER		The format identifier of the transaction identifier
GLOBALID	RAW(64)		The global part (gtrid) of the transaction identifier
BRANCHID	RAW(64)		The branch qualifier (bqual) of the transaction identifier

DBA_PLSQL_OBJECT_SETTINGS

DBA_PLSQL_OBJECT_SETTINGS displays compiler settings for all stored objects in the database. Its columns are the same as those in ALL_PLSQL_OBJECT_SETTINGS.

See Also: ["ALL_PLSQL_OBJECT_SETTINGS"](#) on page 2-84

DBA_POLICIES

DBA_POLICIES describes all security policies in the database. Its columns are the same as those in ALL_POLICIES.

See Also: ["ALL_POLICIES"](#) on page 2-85

DBA_POLICY_CONTEXTS

DBA_POLICY_CONTEXTS describes all driving contexts in the database. Its columns are the same as those in ALL_POLICY_CONTEXTS.

See Also: ["ALL_POLICY_CONTEXTS"](#) on page 2-86

DBA_POLICY_GROUPS

DBA_POLICY_GROUPS describes all policy groups in the database. Its columns are the same as those in ALL_POLICY_GROUPS.

See Also: ["ALL_POLICY_GROUPS"](#) on page 2-86

DBA_PRIV_AUDIT_OPTS

DBA_PRIV_AUDIT_OPTS describes current system privileges being audited across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited. NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client; NULL if the client is performing the operation directly.
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege being audited
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_PROCEDURES

DBA_PROCEDURES lists all functions and procedures along with their associated properties. Its columns are the same as those in "[ALL_PROCEDURES](#)" on page 2-87.

DBA_PROFILES

DBA_PROFILES displays all profiles and their limits.

Column	Datatype	NULL	Description
PROFILE	VARCHAR2(30)	NOT NULL	Profile name
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Resource name
RESOURCE_TYPE	VARCHAR2(8)		Indicates whether the resource profile is a KERNEL or a PASSWORD parameter
LIMIT	VARCHAR2(40)		Limit placed on this resource for this profile

DBA_PROPAGATION

DBA_PROPAGATION displays information about all Streams propagations in the database. Its columns are the same as those in [ALL_PROPAGATION](#).

See Also: "[ALL_PROPAGATION](#)" on page 2-87

DBA_PROXIES

DBA_PROXIES displays Information about all proxy connections in the database.

Related View

USER_PROXIES displays information about connections the current user is allowed to proxy. This view does not display the PROXY column.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of the proxy user
CLIENT	VARCHAR2(30)	NOT NULL	Name of the client user who the proxy user can act on behalf of

Column	Datatype	NULL	Description
AUTHENTICATION	VARCHAR2(3)		Indicates whether the proxy is required to supply the client's authentication credentials (YES) or not (NO)
AUTHORIZATION_ CONSTRAINT	VARCHAR2(35)		Indicates the proxy's authority to exercise roles on the client's behalf: <ul style="list-style-type: none"> ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ NO CLIENT ROLES MAY BE ACTIVATED ■ PROXY MAY ACTIVATE ROLE ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ PROXY MAY NOT ACTIVATE ROLE
ROLE	VARCHAR2(30)		Name of the role referenced in AUTHORIZATION_ CONSTRAINT

See Also: ["USER_PROXIES"](#) on page 3-174

DBA_PUBLISHED_COLUMNS

DBA_PUBLISHED_COLUMNS describes all published source table columns in the database. A user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers. Its columns are the same as those in ALL_PUBLISHED_COLUMNS.

See Also: ["ALL_PUBLISHED_COLUMNS"](#) on page 2-88

DBA_QUEUE_SCHEDULES

DBA_QUEUE_SCHEDULES describes the current schedules for propagating messages.

Related View

USER_QUEUE_SCHEDULES describes current schedules related to the current user. This view does not display the SCHEMA column.

Column	Datatype	NULL	Description
SCHEMA	VARCHAR2(30)	NOT NULL	The schema
QNAME	VARCHAR2(30)	NOT NULL	Source queue name
DESTINATION	VARCHAR2(128)	NOT NULL	Destination name, currently limited to be a DBLINK name
START_DATE	DATE		Date to start propagation in the default date format
START_TIME	VARCHAR2(8)		Time of day at which to start propagation in HH:MI:SS format
PROPAGATION_WINDOW	NUMBER		Duration in seconds for the propagation window
NEXT_TIME	VARCHAR2(200)		Function to compute the start of the next propagation window
LATENCY	NUMBER		Maximum wait time to propagate a message during the propagation window
SCHEDULE_DISABLED	VARCHAR2(1)		N if enabled; Y if disabled and schedule will not be executed
PROCESS_NAME	VARCHAR2(8)		The name of the process executing the schedule; NULL if not currently executing
SESSION_ID	NUMBER		The session ID (SID, SERIAL#) of the job executing this schedule; NULL if not currently executing

Column	Datatype	NULL	Description
INSTANCE	NUMBER		The cluster database instance number executing the schedule
LAST_RUN_DATE	DATE		The date on the last successful execution
LAST_RUN_TIME	VARCHAR2(8)		The time of the last successful execution in HH:MI:SS format
CURRENT_START_DATE	DATE		The date at which the current window of this schedule was started
CURRENT_START_TIME	VARCHAR2(8)		The time of day at which the current window of this schedule was started in HH:MI:SS format
NEXT_RUN_DATE	DATE		The date at which the next window of this schedule will be started
NEXT_RUN_TIME	VARCHAR2(8)		The time of day at which the next window of this schedule will be started in HH:MI:SS format
TOTAL_TIME	NUMBER		The total time, in seconds, spent by the system in executing this schedule
TOTAL_NUMBER	NUMBER		The total number of messages propagated in this schedule
TOTAL_BYTES	NUMBER		The total number of bytes propagated in this schedule
MAX_NUMBER	NUMBER		The maximum number of messages propagated in a propagation window
MAX_BYTES	NUMBER		The maximum number of bytes propagated in a propagation window
AVG_NUMBER	NUMBER		The average number of messages propagated in a propagation window
AVG_SIZE	NUMBER		The average size of a propagated message in bytes
AVG_TIME	NUMBER		The average time, in seconds, to propagate a message
FAILURES	NUMBER		The number of times the execution failed; if 16, the schedule will be disabled
LAST_ERROR_DATE	DATE		The date of the last unsuccessful execution
LAST_ERROR_TIME	VARCHAR2(8)		The time of the last unsuccessful execution
LAST_ERROR_MSG	VARCHAR2(4000)		The error number and error message text of the last unsuccessful execution

DBA_QUEUE_TABLES

DBA_QUEUE_TABLES describes the queues in all queue tables in the database. Its columns are the same as those in ALL_QUEUE_TABLES.

See Also: ["ALL_QUEUE_TABLES"](#) on page 2-88

DBA_QUEUES

DBA_QUEUES describes the operational characteristics of every queue in a database. Its columns are the same as those in "ALL_QUEUES" on page 2-89.

See Also: *Oracle Streams Advanced Queuing User's Guide and Reference* for more information about this view and Advanced Queuing

DBA_RCHILD

DBA_RCHILD lists all the children in any refresh group.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE#	VARCHAR2(30)		Type of the object in the refresh group

DBA_RECYCLEBIN

DBA_RECYCLEBIN displays information about all recycle bins in the database.

Related View

USER_RECYCLEBIN displays information about the recycle bin owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the original owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	New name of the object
ORIGINAL_NAME	VARCHAR2(32)		Original name of the object
OPERATION	VARCHAR2(9)		Operation carried out on the object: <ul style="list-style-type: none"> ▪ DROP - Object was dropped ▪ TRUNCATE - Object was truncated <p>Note: The Oracle Database currently only supports recovering dropped objects from the recycle bin. The truncated objects cannot be recovered.</p>
TYPE	VARCHAR2(25)		Type of the object: <ul style="list-style-type: none"> ▪ TABLE ▪ NORMAL INDEX ▪ BITMAP INDEX ▪ NESTED TABLE ▪ LOB ▪ LOB INDEX ▪ DOMAIN INDEX ▪ IOT TOP INDEX ▪ IOT OVERFLOW SEGMENT ▪ IOT MAPPING TABLE ▪ TRIGGER ▪ CONSTRAINT ▪ Table Partition ▪ Table Composite Partition ▪ Index Partition ▪ Index Composite Partition ▪ LOB Partition ▪ LOB Composite Partition
TS_NAME	VARCHAR2(30)		Name of the tablespace to which the object belongs
CREATETIME	VARCHAR2(19)		Timestamp for the creation of the object
DROPTIME	VARCHAR2(19)		Timestamp for the dropping of the object
DROPSCN	NUMBER		System change number (SCN) of the transaction which moved the object to the recycle bin
PARTITION_NAME	VARCHAR2(32)		Name of the partition which was dropped

Column	Datatype	NULL	Description
CAN_UNDROP	VARCHAR2(3)		Indicates whether the object can be undropped (YES) or not (NO)
CAN_PURGE	VARCHAR2(3)		Indicates whether the object can be purged (YES) or not (NO)
RELATED	NUMBER	NOT NULL	Object number of the parent object
BASE_OBJECT	NUMBER	NOT NULL	Object number of the base object
PURGE_OBJECT	NUMBER	NOT NULL	Object number for the object which gets purged
SPACE	NUMBER		Number of blocks used by the object

See Also: ["USER_RECYCLEBIN"](#) on page 3-175

DBA_REDEFINITION_ERRORS

DBA_REDEFINITION_ERRORS is an online redefinition view and displays the dependent objects for which errors were raised while attempting to create similar objects on the interim table of the redefinition.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(10)		Type of the redefinition object: <ul style="list-style-type: none"> ■ TABLE ■ INDEX ■ CONSTRAINT ■ TRIGGER
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the redefinition object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the redefinition object
BASE_TABLE_OWNER	VARCHAR2(30)		Owner of the base table of the redefinition object
BASE_TABLE_NAME	VARCHAR2(30)		Name of the base table of the redefinition object
DDL_TXT	CLOB		DDL used to create the corresponding interim dependent object

DBA_REDEFINITION_OBJECTS

DBA_REDEFINITION_OBJECTS is an online redefinition view and displays the objects involved in the current redefinitions.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(10)		Type of the redefinition object: <ul style="list-style-type: none"> ■ TABLE ■ INDEX ■ CONSTRAINT ■ TRIGGER
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the redefinition object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the redefinition object
BASE_TABLE_OWNER	VARCHAR2(30)		Owner of the base table of the redefinition object
BASE_TABLE_NAME	VARCHAR2(30)		Name of the base table of the redefinition object
INTERIM_OBJECT_OWNER	VARCHAR2(30)		Owner of the corresponding interim redefinition object
INTERIM_OBJECT_NAME	VARCHAR2(30)		Name of the corresponding interim redefinition object

DBA_REFRESH

DBA_REFRESH describes all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH" on page 2-90.

DBA_REFRESH_CHILDREN

DBA_REFRESH_CHILDREN lists all of the objects in all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 2-91.

DBA_REFS

DBA_REFS describes the REF columns and REF attributes in object type columns of all the objects in the database. Its columns are the same as those in "ALL_REFS" on page 2-92.

DBA_REGISTERED_ARCHIVED_LOG

DBA_REGISTERED_ARCHIVED_LOG displays information about the registered archived logfiles in the database.

Column	Datatype	NULL	Description
CONSUMER_NAME	VARCHAR2(30)	NOT NULL	Consumer name of the archived logs
SOURCE_DATABASE	VARCHAR2(128)		Name of the database which generated the redo logs
THREAD#	NUMBER	NOT NULL	Thread number of the archived redo log. The thread number is 1 for a single instance. For Real Application Clusters, this column will contain different numbers.
SEQUENCE#	NUMBER	NOT NULL	Sequence number of the archived redo log file
FIRST_SCN	NUMBER	NOT NULL	System change number (SCN) of the current archived redo log
NEXT_SCN	NUMBER	NOT NULL	System change number (SCN) of the next archived redo log
FIRST_TIME	DATE		Date and time of the current archived redo log
NEXT_TIME	DATE		Date and time of the next archived redo log
NAME	VARCHAR2(513)		Name of the archived redo log
MODIFIED_TIME	DATE		Time when the archived redo log was registered
DICTIONARY_BEGIN	VARCHAR2(3)		Indicates whether the beginning of the dictionary build is in this archived redo log (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)		Indicates whether the end of the dictionary build is in this archived redo log (YES) or not (NO)

DBA_REGISTERED_MVIEW_GROUPS

DBA_REGISTERED_MVIEW_GROUPS lists all the registered materialized view groups at the master site or master materialized view site.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the materialized view replication group
MVIEW_SITE	VARCHAR2(128)		Site of the materialized view replication group
GROUP_COMMENT	VARCHAR2(80)		Description of the materialized view replication group

Column	Datatype	NULL	Description
VERSION	VARCHAR2(8)		Oracle release of the materialized view replication group: <ul style="list-style-type: none"> ▪ ORACLE 7 ▪ ORACLE 8 ▪ REPAPI Note: Oracle8i and newer materialized view groups show ORACLE 8.
FNAME	VARCHAR2(30)		Name of the flavor of the materialized view group
OWNER	VARCHAR2(30)		Owner of the materialized view replication group

DBA_REGISTERED_MVIEWS

DBA_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) in the database. Its columns are the same as those in ALL_REGISTERED_MVIEWS.

See Also: ["ALL_REGISTERED_MVIEWS"](#) on page 2-92

DBA_REGISTRY

DBA_REGISTRY displays information about the components loaded into the database.

Related View

USER_REGISTRY displays information about the components loaded into the database that are owned by the current user.

Column	Datatype	NULL	Description
COMP_ID	VARCHAR2(30)	NOT NULL	Component identifier
COMP_NAME	VARCHAR2(255)		Component name
VERSION	VARCHAR2(30)		Component version loaded
STATUS	VARCHAR2(11)		Component status: <ul style="list-style-type: none"> ▪ INVALID ▪ VALID ▪ LOADING ▪ LOADED ▪ UPGRADING ▪ UPGRADED ▪ DOWNGRADING ▪ DOWNGRADED ▪ REMOVING ▪ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified
NAMESPACE	VARCHAR2(30)	NOT NULL	Component namespace
CONTROL	VARCHAR2(30)	NOT NULL	User that created the component entry
SCHEMA	VARCHAR2(30)	NOT NULL	User that contains the objects for the component
PROCEDURE	VARCHAR2(61)		Validation procedure
STARTUP	VARCHAR2(8)		Indicates whether the component requires a startup after the upgrade (REQUIRED) or not
PARENT_ID	VARCHAR2(30)		Parent component identifier

See Also: ["USER_REGISTRY"](#) on page 3-176

DBA_REGISTRY_HIERARCHY

DBA_REGISTRY_HIERARCHY displays information about the components loaded into the database, grouped by owner and organized in the component hierarchy.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	Component namespace
COMP_ID	VARCHAR2(4000)		Component identifier
VERSION	VARCHAR2(30)		Component version loaded
STATUS	VARCHAR2(11)		Component status: <ul style="list-style-type: none"> ■ INVALID ■ VALID ■ LOADING ■ LOADED ■ UPGRADING ■ UPGRADED ■ DOWNGRADING ■ DOWNGRADED ■ REMOVING ■ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified

DBA_REPAIR_TABLE

DBA_REPAIR_TABLE describes any corruptions found by the DBMS_REPAIR.CHECK_OBJECT procedure. This information is used by the DBMS_REPAIR.FIX_CORRUPT_BLOCKS procedure on execution. To create this view, first run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the resulting repair table for an object, run the DBMS_REPAIR.CHECK_OBJECT procedure on the object.

Note: The table created by the DBMS_REPAIR.ADMIN_TABLES procedure is called REPAIR_TABLE by default. If you specify a different name, this view will have the name you specify, preceded by "DBA_REPAIR_".

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object with the corruption
TABLESPACE_ID	NUMBER	NOT NULL	Tablespace number of the corrupt object
RELATIVE_FILE_ID	NUMBER)	NOT NULL	Relative file number of the corrupt object
BLOCK_ID	NUMBER	NOT NULL	Block number of the corruption
CORRUPT_TYPE	NUMBER	NOT NULL	Type of corruption encountered
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema of the corrupt object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the corrupt object
BASEOBJECT_NAME	VARCHAR2(30)	NULL	If the object is an index, the name of its base table
PARTITION_NAME	VARCHAR2(30)	NULL	Partition or subpartition name, if applicable

Column	Datatype	NULL	Description
CORRUPT_DESCRIPTION	VARCHAR2(200)	NULL	Description of corruption
REPAIR_DESCRIPTION	VARCHAR2(200)	NULL	Description of repair action
MARKED_CORRUPT	VARCHAR2(10)	NOT NULL	Whether the block is marked corrupt (TRUE FALSE)
CHECK_TIMESTAMP	DATE	NOT NULL	Date and time when this row was insert into the repair table
FIX_TIMESTAMP	DATE	NULL	Date and time when the block was modified by the FIX_CORRUPT_BLOCKS procedure, if applicable
REFORMAT_TIMESTAMP	DATE	NULL	Reserved for future use

DBA_RESUMABLE

DBA_RESUMABLE lists all resumable statements executed in the system.

Related View

USER_RESUMABLE lists resumable statements executed by the current user. This view does not display the USER_ID column.

Column	Datatype	NULL	Description
USER_ID	NUMBER		User ID Number of the Resumable Statement Owner
SESSION_ID	NUMBER		Session Identifier of the Resumable Statement
INSTANCE_ID	NUMBER		Instance Number of the Resumable Statement
COORD_INSTANCE_ID	NUMBER		Instance Number on which the Parallel Coordinator is Running
COORD_SESSION_ID	NUMBER		Session Identifier of the Parallel Coordinator
STATUS	VARCHAR2(9)		Status of the resumable statement: <ul style="list-style-type: none"> ■ RUNNING ■ SUSPENDED ■ TIMEOUT ■ ERROR ■ ABORTED
TIMEOUT	NUMBER		Timeout of the resumable statement
START_TIME	VARCHAR2(20)		Start time of the resumable statement
SUSPEND_TIME	VARCHAR2(20)		Last time the resumable statement was suspended. It is initialized to null.
RESUME_TIME	VARCHAR2(20)		Last time the suspended resumable statement was resumed. It is initialized to null.
NAME	VARCHAR2(4000)		Name given in the resumable clause of the resumable statement
SQL_TEXT	VARCHAR2(1000)		Resumable statement, selected from the V\$SQL view
ERROR_NUMBER	NUMBER		Error code of the last correctable error. When STATUS is set to RUNNING, its value will be set to 0.
ERROR_PARAMETER1	VARCHAR2(80)		First parameter for the error message (null if no error)
ERROR_PARAMETER2	VARCHAR2(80)		Second parameter for the error message (null if no error)
ERROR_PARAMETER3	VARCHAR2(80)		Third parameter for the error message (null if no error)
ERROR_PARAMETER4	VARCHAR2(80)		Forth parameter for the error message (null if no error)
ERROR_PARAMETER5	VARCHAR2(80)		Fifth parameter for the error message (null if no error)
ERROR_MSG	VARCHAR2(4000)		Error message corresponding to ERROR_NUMBER. It will be null when ERROR_NUMBER is null.

See Also: ["USER_RESUMABLE"](#) on page 3-176

DBA_REWRITE_EQUIVALENCES

DBA_REWRITE_EQUIVALENCES describes all rewrite equivalences in the database. Its columns are the same as those in ALL_REWRITE_EQUIVALENCES.

See Also: ["ALL_REWRITE_EQUIVALENCES"](#) on page 2-93

DBA_RGROUP

DBA_RGROUP lists all refresh groups.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		(Y or N) If Y, then destroy the refresh group when its last item is removed
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y or N) If Y, then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		(Y or N) If Y, proceed with refresh despite error when pushing deferred RPC's
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER	NOT NULL	Identifier of job used to refresh the group automatically
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

DBA_ROLE_PRIVS

DBA_ROLE_PRIVS describes the roles granted to all users and roles in the database.

Related View

USER_ROLE_PRIVS describes the roles granted to the current user.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)		Name of the user or role receiving the grant
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Granted role name
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN OPTION (YES) or not (NO)
DEFAULT_ROLE	VARCHAR2(3)		Indicates whether the role is designated as a DEFAULT ROLE for the user (YES) or not (NO)

See Also: ["USER_ROLE_PRIVS"](#) on page 3-176

DBA_ROLES

DBA_ROLES lists all roles that exist in the database.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Role name
PASSWORD_REQUIRED	VARCHAR2(8)		Indicates if the role requires a password to be enabled

DBA_ROLLBACK_SEGS

DBA_ROLLBACK_SEGS describes rollback segments.

Column	Datatype	NULL	Description
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the rollback segment
OWNER	VARCHAR2(6)		Owner of the rollback segment
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the rollback segment
SEGMENT_ID	NUMBER	NOT NULL	ID number of the rollback segment
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the segment head
BLOCK_ID	NUMBER	NOT NULL	ID number of the block containing the segment header
INITIAL_EXTENT	NUMBER		Initial extent size in bytes
NEXT_EXTENT	NUMBER		Secondary extent size in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extent
PCT_INCREASE	NUMBER		Percent increase for extent size
STATUS	VARCHAR2(16)		Rollback segment status
INSTANCE_NUM	VARCHAR2(40)		Rollback segment owning Real Application Clusters instance number
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the segment header

DBA_RSRC_CONSUMER_GROUP_PRIVS

DBA_RSRC_CONSUMER_GROUP_PRIVS displays information about all resource consumer groups and the users and roles assigned to them. The grant referred to in this view and the related view is the grant of the SWITCH_CONSUMER_GROUP object privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. This privilege is not granted through the GRANT SQL statement.

Related View

USER_RSRC_CONSUMER_GROUP_PRIVS displays information about the resource consumer groups to which the current user is assigned. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
GRANTED_GROUP	VARCHAR2(30)		Granted consumer group name
GRANT_OPTION	VARCHAR2(3)		Indicates whether the grant was with the GRANT option (YES) or not (NO)
INITIAL_GROUP	VARCHAR2(3)		Indicates whether the consumer group is designated as the default for this user or role (YES) or not (NO)

See Also:

- ["USER_RSRC_CONSUMER_GROUP_PRIVS"](#) on page 3-177
- *PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_CONSUMER_GROUPS

DBA_RSRC_CONSUMER_GROUPS displays information about all resource consumer groups in the database.

Column	Datatype	NULL	Description
CONSUMER_GROUP	VARCHAR2(30)		Name of the consumer group
CPU_METHOD	VARCHAR2(30)		CPU resource allocation method for the consumer group
COMMENTS	VARCHAR2(2000)		Text comment on the consumer group
STATUS	VARCHAR2(30)		Status of the consumer group: <ul style="list-style-type: none"> ▪ PENDING - Consumer group is part of the pending area ▪ ACTIVE
MANDATORY	VARCHAR2(3)		Indicates whether the consumer group is mandatory (YES) or not (NO)

DBA_RSRC_GROUP_MAPPINGS

DBA_RSRC_GROUP_MAPPINGS displays the mapping between session attributes and consumer groups in the database.

Column	Datatype	NULL	Description
ATTRIBUTE	VARCHAR2(30)		Session attribute to match
VALUE	VARCHAR2(128)		Attribute value
CONSUMER_GROUP	VARCHAR2(30)		Target consumer group name
STATUS	VARCHAR2(30)		Status of the consumer group: <ul style="list-style-type: none"> ▪ PENDING - Consumer group is part of the pending area ▪ ACTIVE

DBA_RSRC_MANAGER_SYSTEM_PRIVS

DBA_RSRC_MANAGER_SYSTEM_PRIVS displays information about all the users and roles that have been granted the ADMINISTER_RESOURCE_MANAGER system privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. This privilege is not granted through the GRANT SQL statement.

Related View

USER_RSRC_MANAGER_SYSTEM_PRIVS displays information about the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege

Column	Datatype	NULL	Description
ADMIN_OPTION	VARCHAR2 (3)		Indicates whether the grant was with the ADMIN option (YES) or not (NO)

See Also:

- ["USER_RSRC_MANAGER_SYSTEM_PRIVS"](#) on page 3-177
- *PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_MAPPING_PRIORITY

DBA_RSRC_MAPPING_PRIORITY displays information about all consumer group mapping attribute priorities.

Column	Datatype	NULL	Description
ATTRIBUTE	VARCHAR2 (30)		Session attribute
PRIORITY	NUMBER		Priority (1 is the highest)
STATUS	VARCHAR2 (30)		Status of the consumer group: <ul style="list-style-type: none"> ■ PENDING - Consumer group is part of the pending area ■ ACTIVE

DBA_RSRC_PLAN_DIRECTIVES

DBA_RSRC_PLAN_DIRECTIVES displays information about all resource plan directives in the database.

Column	Datatype	Description
PLAN	VARCHAR2 (30)	Name of the plan to which the directive belongs
GROUP_OR_SUBPLAN	VARCHAR2 (30)	Name of the consumer group or subplan referred to
TYPE	VARCHAR2 (14)	Indicates whether GROUP_OR_SUBPLAN refers to a consumer group (CONSUMER_GROUP) or a plan (PLAN)
CPU_P1	NUMBER	First parameter for the CPU resource allocation method
CPU_P2	NUMBER	Second parameter for the CPU resource allocation method
CPU_P3	NUMBER	Third parameter for the CPU resource allocation method
CPU_P4	NUMBER	Fourth parameter for the CPU resource allocation method
CPU_P5	NUMBER	Fifth parameter for the CPU resource allocation method
CPU_P6	NUMBER	Sixth parameter for the CPU resource allocation method
CPU_P7	NUMBER	Seventh parameter for the CPU resource allocation method
CPU_P8	NUMBER	Eighth parameter for the CPU resource allocation method
ACTIVE_SESS_POOL_P1	NUMBER	First parameter for the active session pool resource allocation method
QUEUEING_P1	NUMBER	First parameter for the queuing resource allocation method
PARALLEL_DEGREE_LIMIT_P1	NUMBER	First parameter for the parallel degree limit resource allocation method
SWITCH_GROUP	VARCHAR2 (30)	Group to switch to once the switch time is reached
SWITCH_TIME	NUMBER	Amount of run time before the session is automatically switched
SWITCH_ESTIMATE	VARCHAR2 (5)	Indicates whether estimated execution time should be used for switch criteria (TRUE) or not (FALSE)

Column	Datatype	Description
MAX_EST_EXEC_TIME	NUMBER	Maximum estimated execution time
UNDO_POOL	NUMBER	Undo pool size for the consumer group
MAX_IDLE_TIME	NUMBER	Maximum idle time for the session
MAX_IDLE_BLOCKER_TIME	NUMBER	Maximum idle time for the session when blocking other sessions
SWITCH_TIME_IN_CALL	NUMBER	Call switch time limit for execution in a group
COMMENTS	VARCHAR2(2000)	Text comment on the plan directive
STATUS	VARCHAR2(30)	Status of the directive: <ul style="list-style-type: none"> ■ PENDING - Directive is part of the pending area ■ ACTIVE
MANDATORY	VARCHAR2(3)	Indicates whether the plan directive is mandatory (YES) or not (NO)

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans in general
- *PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

DBA_RSRC_PLANS

DBA_RSRC_PLANS displays information about all resource plans in the database. For a list of currently active plans, see "[V\\$RSRC_PLAN](#)" on page 5-29.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		Name of the resource plan
NUM_PLAN_DIRECTIVES	NUMBER		Number of plan directives for the plan
CPU_METHOD	VARCHAR2(30)		CPU resource allocation method for the plan
ACTIVE_SESS_POOL_MTH	VARCHAR2(30)		Active session pool resource allocation method for the plan
PARALLEL_DEGREE_LIMIT_MTH	VARCHAR2(30)		Parallel degree limit resource allocation method for the plan
QUEUING_MTH	VARCHAR2(30)		Queuing resource allocation method for the plan
COMMENTS	VARCHAR2(2000)		Text comment on the plan
STATUS	VARCHAR2(30)		Status of the plan: <ul style="list-style-type: none"> ■ PENDING - Plan is part of the pending area ■ ACTIVE
MANDATORY	VARCHAR2(3)		Indicates whether the plan is mandatory (YES) or not (NO)

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans in general
- *PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

DBA_RULE_SET_RULES

DBA_RULE_SET_RULES describes the rules in all rule sets in the database. Its columns are the same as those in ALL_RULE_SET_RULES.

See Also: ["ALL_RULE_SET_RULES"](#) on page 2-93

DBA_RULE_SETS

DBA_RULE_SETS describes all rule sets in the database. Its columns are the same as those in ALL_RULE_SETS.

See Also: ["ALL_RULE_SETS"](#) on page 2-94

DBA_RULES

DBA_RULES describes all rules in the database. Its columns are the same as those in ALL_RULES.

See Also: ["ALL_RULES"](#) on page 2-94

DBA_SCHEDULER_JOB_ARGS

DBA_SCHEDULER_JOB_ARGS displays information about the arguments of all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_ARGS.

See Also: ["ALL_SCHEDULER_JOB_ARGS"](#) on page 2-95

DBA_SCHEDULER_JOB_CLASSES

DBA_SCHEDULER_JOB_CLASSES displays information about all Scheduler job classes in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_CLASSES.

See Also: ["ALL_SCHEDULER_JOB_CLASSES"](#) on page 2-96

DBA_SCHEDULER_JOB_LOG

DBA_SCHEDULER_JOB_LOG displays log information for all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_LOG.

See Also: ["ALL_SCHEDULER_JOB_LOG"](#) on page 2-96

DBA_SCHEDULER_JOB_RUN_DETAILS

DBA_SCHEDULER_JOB_RUN_DETAILS displays log run details for all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_RUN_DETAILS.

See Also: ["ALL_SCHEDULER_JOB_RUN_DETAILS"](#) on page 2-97

DBA_SCHEDULER_JOBS

DBA_SCHEDULER_JOBS displays information about all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOBS.

See Also: ["ALL_SCHEDULER_JOBS"](#) on page 2-97

DBA_SCHEDULER_PROGRAM_ARGS

DBA_SCHEDULER_PROGRAM_ARGS displays information about the arguments of all Scheduler programs in the database. Its columns are the same as those in ALL_SCHEDULER_PROGRAM_ARGS.

See Also: ["ALL_SCHEDULER_PROGRAM_ARGS"](#) on page 2-99

DBA_SCHEDULER_PROGRAMS

DBA_SCHEDULER_PROGRAMS displays information about all Scheduler programs in the database. Its columns are the same as those in ALL_SCHEDULER_PROGRAMS.

See Also: ["ALL_SCHEDULER_PROGRAMS"](#) on page 2-100

DBA_SCHEDULER_RUNNING_JOBS

DBA_SCHEDULER_RUNNING_JOBS displays information about all running Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_RUNNING_JOBS.

See Also: ["ALL_SCHEDULER_RUNNING_JOBS"](#) on page 2-101

DBA_SCHEDULER_SCHEDULES

DBA_SCHEDULER_SCHEDULES displays information about all Scheduler schedules in the database. Its columns are the same as those in ALL_SCHEDULER_SCHEDULES.

See Also: ["ALL_SCHEDULER_SCHEDULES"](#) on page 2-101

DBA_SCHEDULER_WINDOW_DETAILS

DBA_SCHEDULER_WINDOW_DETAILS displays log details for all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_DETAILS.

See Also: ["ALL_SCHEDULER_WINDOW_DETAILS"](#) on page 2-102

DBA_SCHEDULER_WINDOW_GROUPS

DBA_SCHEDULER_WINDOW_GROUPS displays information about all Scheduler window groups in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_GROUPS.

See Also: ["ALL_SCHEDULER_WINDOW_GROUPS"](#) on page 2-102

DBA_SCHEDULER_WINDOW_LOG

DBA_SCHEDULER_WINDOW_LOG displays log information for all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_LOG.

See Also: ["ALL_SCHEDULER_WINDOW_LOG"](#) on page 2-103

DBA_SCHEDULER_WINDOWS

DBA_SCHEDULER_WINDOWS displays information about all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOWS.

See Also: ["ALL_SCHEDULER_WINDOWS"](#) on page 2-103

DBA_SCHEDULER_WINGROUP_MEMBERS

DBA_SCHEDULER_WINGROUP_MEMBERS displays the members of all Scheduler window groups in the database. Its columns are the same as those in ALL_SCHEDULER_WINGROUP_MEMBERS.

See Also: ["ALL_SCHEDULER_WINGROUP_MEMBERS"](#) on page 2-104

DBA_SEC_RELEVANT_COLS

DBA_SEC_RELEVANT_COLS describes the security relevant columns of all security policies in the database. Its columns are the same as those in ALL_SEC_RELEVANT_COLS.

See Also: ["ALL_SEC_RELEVANT_COLS"](#) on page 2-104

DBA_SEGMENTS

DBA_SEGMENTS describes the storage allocated for all segments in the database.

Related View

USER_SEGMENTS describes the storage allocated for the segments owned by the current user's objects. This view does not display the OWNER, HEADER_FILE, HEADER_BLOCK, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the segment owner
SEGMENT_NAME	VARCHAR2(81)		Name, if any, of the segment
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(18)		Type of segment: INDEX PARTITION, TABLE PARTITION, TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE, LOBSEGMENT and LOBINDEX
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
BYTES	NUMBER		Size in bytes, of the segment
BLOCKS	NUMBER		Size, in Oracle blocks, of the segment
EXTENTS	NUMBER		Number of extents allocated to the segment
INITIAL_EXTENT	NUMBER		Size in bytes requested for the initial extent of the segment at create time. (Oracle rounds the extent size to multiples of 5 blocks if the requested size is greater than 5 blocks.)
NEXT_EXTENT	NUMBER		Size in bytes of the next extent to be allocated to the segment

Column	Datatype	NULL	Description
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percent by which to increase the size of the next extent to be allocated
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
RELATIVE_FNO	NUMBER		Relative file number of the segment header
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the object

See Also: ["USER_SEGMENTS"](#) on page 3-179

DBA_SEQUENCES

DBA_SEQUENCES describes all sequences in the database. Its columns are the same as those in ["ALL_SEQUENCES"](#) on page 2-105.

DBA_SERVER_REGISTRY

DBA_SERVER_REGISTRY displays information about the components loaded into the database.

Column	Datatype	NULL	Description
COMP_ID	VARCHAR2(30)	NOT NULL	Component identifier
COMP_NAME	VARCHAR2(255)		Component name
VERSION	VARCHAR2(30)		Component version loaded
STATUS	VARCHAR2(11)		Component status: <ul style="list-style-type: none"> ▪ INVALID ▪ VALID ▪ LOADING ▪ LOADED ▪ UPGRADING ▪ UPGRADED ▪ DOWNGRADING ▪ DOWNGRADED ▪ REMOVING ▪ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified
CONTROL	VARCHAR2(30)	NOT NULL	User that created the component entry
SCHEMA	VARCHAR2(30)	NOT NULL	User that contains the objects for the component
PROCEDURE	VARCHAR2(61)		Validation procedure
STARTUP	VARCHAR2(8)		Indicates whether the component requires a startup after the upgrade (REQUIRED) or not
PARENT_ID	VARCHAR2(30)		Parent component identifier

DBA_SERVICES

DBA_SERVICES displays all services in the database. The view excludes rows marked for deletion. Its columns are the same as those in [ALL_SERVICES](#).

DBA_SOURCE

DBA_SOURCE describes the text source of all stored objects in the database. Its columns are the same as those in ["ALL_SOURCE"](#) on page 2-106.

DBA_SOURCE_TABLES

DBA_SOURCE_TABLES describes all existing source tables in the database. A user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers. Its columns are the same as those in ALL_SOURCE_TABLES.

See Also: ["ALL_SOURCE_TABLES"](#) on page 2-106

DBA_SQL_PROFILES

DBA_SQL_PROFILES displays information about SQL profiles currently created for specific SQL statements.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the SQL profile
CATEGORY	VARCHAR2(30)	NOT NULL	Category of the SQL profile
SIGNATURE	NUMBER	NOT NULL	Unique identifier generated from normalized SQL text
SQL_TEXT	CLOB	NOT NULL	Un-normalized SQL text
CREATED	DATE	NOT NULL	Timestamp when the SQL profile was created
LAST_MODIFIED	DATE	NOT NULL	Timestamp when the SQL profile was last modified
DESCRIPTION	VARCHAR2(500)		Text description provided for the SQL profile
TYPE	VARCHAR2(9)		Type of the SQL profile (how it was created): <ul style="list-style-type: none"> ▪ MANUAL ▪ AUTO-TUNE
STATUS	VARCHAR2(8)		Status of the SQL profile: <ul style="list-style-type: none"> ▪ ENABLED ▪ DISABLED ▪ VOID

See Also: The DBMS_SQLTUNE package in *PL/SQL Packages and Types Reference*

DBA_SQLJ_TYPE_ATTRS

DBA_SQLJ_TYPE_ATTRS describes the attributes of all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPE_ATTRS.

See Also: ["ALL_SQLJ_TYPE_ATTRS"](#) on page 2-107

DBA_SQLJ_TYPE_METHODS

DBA_SQLJ_TYPE_METHODS describes the methods of all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPE_METHODS.

See Also: ["ALL_SQLJ_TYPE_METHODS"](#) on page 2-107

DBA_SQLJ_TYPES

DBA_SQLJ_TYPES describes all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPES.

See Also: ["ALL_SQLJ_TYPES"](#) on page 2-108

DBA_SQLSET

DBA_SQLSET displays information about all SQL tuning sets in the database.

Related View

USER_SQLSET displays information about the SQL tuning sets owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	SQL tuning set identifier
NAME	VARCHAR2(30)	NOT NULL	Name of the SQL tuning set
OWNER	VARCHAR2(30)		Owner of the SQL tuning set
DESCRIPTION	VARCHAR2(256)		Description of the SQL tuning set
CREATED	DATE		Date of creation of the SQL tuning set
LAST_MODIFIED	DATE		Date of the last modification of the SQL tuning set
STATEMENT_COUNT	NUMBER		Number of statements in the SQL tuning set

See Also: ["USER_SQLSET"](#) on page 3-179

DBA_SQLSET_BINDS

DBA_SQLSET_BINDS displays the bind values associated with all SQL tuning sets in the database.

Related View

USER_SQLSET_BINDS displays the bind values associated with the SQL tuning sets owned by the current user.

Column	Datatype	NULL	Description
SQLSET_ID	NUMBER	NOT NULL	SQL tuning set identifier
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier
POSITION	NUMBER	NOT NULL	Bind position
VALUE	ANYDATA		Bind value

See Also: ["USER_SQLSET_BINDS"](#) on page 3-180

DBA_SQLSET_REFERENCES

DBA_SQLSET_REFERENCES describes whether or not all SQL tuning sets in the database are active. A SQL tuning set cannot be dropped if it is referenced.

Related View

USER_SQLSET_REFERENCES describes whether or not the SQL tuning sets owned by the current user are active.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Reference identifier
SQLSET_ID	NUMBER	NOT NULL	SQL tuning set identifier
OWNER	VARCHAR2(30)		User who registered to use the SQL tuning set
CREATED	DATE		Date the reference was created
DESCRIPTION	VARCHAR2(256)		Description of the usage of the SQL tuning set

See Also: ["USER_SQLSET_REFERENCES"](#) on page 3-180

DBA_SQLSET_STATEMENTS

DBA_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form all SQL tuning sets in the database.

Related View

USER_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form the SQL tuning sets owned by the current user.

Column	Datatype	NULL	Description
SQLSET_ID	NUMBER	NOT NULL	SQL tuning set identifier
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the SQL statement
SQL_TEXT	CLOB		Text of the SQL statement
PARSING_SCHEMA_ID	NUMBER		Schema under which the SQL is parsed
MODULE	VARCHAR2(48)		Last application module recorded for the SQL
ACTION	VARCHAR2(32)		Last application action recorded for the SQL
ELAPSED_TIME	NUMBER		Elapsed time for the SQL statement
CPU_TIME	NUMBER		CPU time for the SQL
BUFFER_GETS	NUMBER		Number of buffer gets
DISK_READS	NUMBER		Number of disk reads
ROWS_PROCESSED	NUMBER		Number of rows processed by the SQL
FETCHES	NUMBER		Number of fetches
EXECUTIONS	NUMBER		Number of executions
END_OF_FETCH_COUNT	NUMBER		End of fetch count
OPTIMIZER_COST	NUMBER		Optimizer cost for the SQL
OPTIMIZER_ENV	RAW(1000)		Optimizer environment
PRIORITY	NUMBER		User-defined priority
COMMAND_TYPE	NUMBER		Command type
STAT_PERIOD	NUMBER		Period of time (in seconds) during which the statistics of the SQL statement were collected
ACTIVE_STAT_PERIOD	NUMBER		Effective period of time (in seconds) during which the SQL statement was active

See Also: ["USER_SQLSET_STATEMENTS"](#) on page 3-180

DBA_SQLTUNE_BINDS

DBA_SQLTUNE_BINDS displays the bind values associated with all tuned SQL statements in the database.

Related View

USER_SQLTUNE_BINDS displays the bind values associated with the tuned SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
POSITION	NUMBER(38)	NOT NULL	Bind position
VALUE	ANYDATA		Bind value

See Also: ["USER_SQLTUNE_BINDS"](#) on page 3-180

DBA_SQLTUNE_PLANS

DBA_SQLTUNE_PLANS displays information about the execution plans generated for all SQL statements in the database during a SQL tuning session.

Related View

USER_SQLTUNE_PLANS displays information about the execution plans generated for the SQL statements owned by the current user during a SQL tuning session.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
ATTRIBUTE	VARCHAR2(27)		Type of the execution plan: <ul style="list-style-type: none"> ■ Original - Original plan of the query ■ Original with adjusted cost - Same as Original but with adjusted cost ■ Using SQL profile - Plan with sql profile applied ■ Using new indices - Plan with indexes applied
STATEMENT_ID	VARCHAR2(30)		Optional statement identifier specified in the EXPLAIN PLAN statement
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the execution plan
PLAN_ID	NUMBER		Plan identifier
TIMESTAMP	DATE		Date and time when the EXPLAIN PLAN statement was issued
REMARKS	VARCHAR2(4000)		Place for comments that can be added to the steps of the execution plan
OPERATION	VARCHAR2(30)		Name of the operation performed at this step
OPTIONS	VARCHAR2(255)		Options used for the operation performed at this step
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_ALIAS	VARCHAR2(65)		Object alias

Column	Datatype	NULL	Description
OBJECT_INSTANCE	NUMBER (38)		Numbered position of the object name in the original SQL statement
OBJECT_TYPE	VARCHAR2 (30)		Descriptive modifier that further describes the type of object
OPTIMIZER	VARCHAR2 (255)		Current mode of the optimizer
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
ID	NUMBER (38)	NOT NULL	Identification number for this step in the execution plan
PARENT_ID	NUMBER (38)		ID of the next step that operates on the results of this step
DEPTH	NUMBER (38)		Depth
POSITION	NUMBER (38)		Order of processing for steps with the same parent ID
COST	NUMBER (38)		Cost of the current operation estimated by the cost-based optimizer (CBO)
CARDINALITY	NUMBER (38)		Number of rows returned by the current operation (estimated by the CBO)
BYTES	NUMBER (38)		Number of bytes returned by the current operation
OTHER_TAG	VARCHAR2 (255)		Describes the function of the SQL text in the OTHER column. Values for OTHER_TAG are: <ul style="list-style-type: none"> ▪ SERIAL - SQL is the text of a locally-executed, serial query plan. Currently, SQL is not loaded in OTHER for this case. ▪ SERIAL_FROM_REMOTE - SQL text shown in the OTHER column will be executed at a remote site ▪ PARALLEL_COMBINED_WITH_PARENT - Parent of this operation is a DFO that performs both operations in the parallel execution plan ▪ PARALLEL_COMBINED_WITH_CHILD - Child of this operation is a DFO that performs both operations in the parallel execution plan. ▪ PARALLEL_TO_SERIAL - SQL text shown in the OTHER column is the top-level of the parallel plan. ▪ PARALLEL_TO_PARALLEL - SQL text shown in the OTHER column is executed and output in parallel ▪ PARALLEL_FROM_SERIAL - Operation consumes data from a serial operation and outputs it in parallel
PARTITION_START	VARCHAR2 (255)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2 (255)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER (38)		Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Information about parallel execution servers and parallel queries
DISTRIBUTION	VARCHAR2 (30)		Distribution method
CPU_COST	NUMBER (38)		User-defined CPU cost
IO_COST	NUMBER (38)		User-defined I/O cost
TEMP_SPACE	NUMBER (38)		Temporary space usage of the operation (sort or hash-join) as estimated by the CBO
ACCESS_PREDICATES	VARCHAR2 (4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2 (4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2 (4000)		Expressions produced by the operation

Column	Datatype	NULL	Description
TIME	NUMBER (38)		Elapsed time (in seconds) of the operation as estimated by the CBO
QBLOCK_NAME	VARCHAR2 (30)		Name of the query block

See Also: ["USER_SQLTUNE_PLANS"](#) on page 3-180

DBA_SQLTUNE_RATIONALE_PLAN

DBA_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of all SQL statements in the database.

Related View

USER_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of the SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER (38)	NOT NULL	Tuning task identifier
RATIONALE_ID	NUMBER (38)	NOT NULL	Rationale identifier
OBJECT_ID	NUMBER (38)	NOT NULL	Advisor framework object identifier
OPERATION_ID	NUMBER (38)	NOT NULL	Operation identifier
PLAN_ATTRIBUTE	VARCHAR2 (27)		Type of the execution plan: <ul style="list-style-type: none"> ▪ Original - Original plan of the query ▪ Original with adjusted cost - Same as Original but with adjusted cost ▪ Using SQL profile - Plan with sql profile applied ▪ Using new indices - Plan with indexes applied

See Also: ["USER_SQLTUNE_RATIONALE_PLAN"](#) on page 3-180

DBA_SQLTUNE_STATISTICS

DBA_SQLTUNE_STATISTICS displays statistics associated with all SQL statements in the database.

Related View

USER_SQLTUNE_STATISTICS displays statistics associated with the SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER (38)	NOT NULL	Tuning task identifier
OBJECT_ID	NUMBER (38)	NOT NULL	Advisor framework object identifier
PARSING_SCHEMA_ID	NUMBER		Schema under which the SQL is parsed
MODULE	VARCHAR2 (48)		Last application module recorded for the SQL
ACTION	VARCHAR2 (32)		Last application action recorded for the SQL
ELAPSED_TIME	NUMBER		Elapsed time for the SQL statement
CPU_TIME	NUMBER		CPU time for the SQL
BUFFER_GETS	NUMBER		Number of buffer gets

Column	Datatype	NULL	Description
DISK_READS	NUMBER		Number of disk reads
ROWS_PROCESSED	NUMBER		Number of rows processed by the SQL
FETCHES	NUMBER		Number of fetches
EXECUTIONS	NUMBER		Number of executions
END_OF_FETCH_COUNT	NUMBER		End of fetch count
OPTIMIZER_COST	NUMBER		Optimizer cost for the SQL
OPTIMIZER_ENV	RAW(1000)		Optimizer environment
COMMAND_TYPE	NUMBER		Command type

See Also: ["USER_SQLTUNE_STATISTICS"](#) on page 3-180

DBA_STMT_AUDIT_OPTS

DBA_STMT_AUDIT_OPTS describes current system auditing options across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited. NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client. NULL if the client is performing the operation directly.
AUDIT_OPTION	VARCHAR2(40)	NOT NULL	Name of the system auditing option
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_STORED_SETTINGS

DBA_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges. It also returns parameter information for all objects in the database and is accessible only to users with the SELECT_CATALOG_ROLE privilege. Its columns are the same as those in ["ALL_STORED_SETTINGS"](#) on page 2-109.

DBA_STREAMS_ADMINISTRATOR

DBA_STREAMS_ADMINISTRATOR displays information about the users who have been granted privileges to be a Streams administrator by procedures in the DBMS_STREAMS_AUTH package.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user who has been granted privileges to be a Streams administrator
LOCAL_PRIVILEGES	VARCHAR2(3)		Indicates whether the user has been granted local Streams administrator privileges (YES) or not (NO)
ACCESS_FROM_REMOTE	VARCHAR2(3)		Indicates whether the user can be used for remote Streams administration through a database link (YES) or not (NO)

DBA_STREAMS_GLOBAL_RULES

DBA_STREAMS_GLOBAL_RULES displays information about the global rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_GLOBAL_RULES.

See Also: ["ALL_STREAMS_GLOBAL_RULES"](#) on page 2-110

DBA_STREAMS_MESSAGE_CONSUMERS

DBA_STREAMS_MESSAGE_CONSUMERS displays information about all Streams messaging clients in the database. Its columns are the same as those in ALL_STREAMS_MESSAGE_CONSUMERS.

See Also: ["ALL_STREAMS_MESSAGE_CONSUMERS"](#) on page 2-110

DBA_STREAMS_MESSAGE_RULES

DBA_STREAMS_MESSAGE_RULES displays information about all Streams messaging rules in the database. Its columns are the same as those in ALL_STREAMS_MESSAGE_RULES.

See Also: ["ALL_STREAMS_MESSAGE_RULES"](#) on page 2-111

DBA_STREAMS_NEWLY_SUPPORTED

DBA_STREAMS_NEWLY_SUPPORTED displays information about all tables in the database that are newly supported by Streams. Its columns are the same as those in ALL_STREAMS_NEWLY_SUPPORTED.

See Also: ["ALL_STREAMS_NEWLY_SUPPORTED"](#) on page 2-111

DBA_STREAMS_RULES

DBA_STREAMS_RULES displays information about the rules used by all Streams processes in the database. Its columns are the same as those in ALL_STREAMS_RULES.

See Also: ["ALL_STREAMS_RULES"](#) on page 2-112

DBA_STREAMS_SCHEMA_RULES

DBA_STREAMS_SCHEMA_RULES displays information about the schema rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_SCHEMA_RULES.

See Also: ["ALL_STREAMS_SCHEMA_RULES"](#) on page 2-113

DBA_STREAMS_TABLE_RULES

DBA_STREAMS_TABLE_RULES displays information about the table rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_TABLE_RULES.

See Also: ["ALL_STREAMS_TABLE_RULES"](#) on page 2-114

DBA_STREAMS_TRANSFORM_FUNCTION

DBA_STREAMS_TRANSFORM_FUNCTION displays information about all rule-based transformation functions in the database. Its columns are the same as those in ALL_STREAMS_TRANSFORM_FUNCTION.

See Also: "[ALL_STREAMS_TRANSFORM_FUNCTION](#)" on page 2-115

DBA_STREAMS_UNSUPPORTED

DBA_STREAMS_UNSUPPORTED displays information about all tables in the database that are not supported by Streams in this release of Oracle. Its columns are the same as those in ALL_STREAMS_UNSUPPORTED.

See Also: "[ALL_STREAMS_UNSUPPORTED](#)" on page 2-116

DBA_SUBPART_COL_STATISTICS

DBA_SUBPART_COL_STATISTICS provides column statistics and histogram information for all subpartitions in the database. Its columns are the same as those in "[ALL_STORED_SETTINGS](#)" on page 2-109.

DBA_SUBPART_HISTOGRAMS

DBA_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on all table subpartitions in the database. Its columns are the same as those in "[ALL_SUBPART_HISTOGRAMS](#)" on page 2-117.

DBA_SUBPART_KEY_COLUMNS

DBA_SUBPART_KEY_COLUMNS lists subpartitioning key columns for all composite-partitioned tables (and local indexes on composite-partitioned tables) in the database. Its columns are the same as those in ALL_SUBPART_KEY_COLUMNS.

See Also: "[ALL_SUBPART_KEY_COLUMNS](#)" on page 2-118

DBA_SUBPARTITION_TEMPLATES

DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database. Its columns are the same as those in ALL_SUBPARTITION_TEMPLATES.

See Also: "[ALL_SUBPARTITION_TEMPLATES](#)" on page 2-118

DBA_SUBSCRIBED_COLUMNS

DBA_SUBSCRIBED_COLUMNS describes the columns of source tables to which any subscriber has subscribed. A user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers. Its columns are the same as those in ALL_SUBSCRIBED_COLUMNS.

See Also: "[ALL_SUBSCRIBED_COLUMNS](#)" on page 2-119

DBA_SUBSCRIBED_TABLES

DBA_SUBSCRIBED_TABLES describes all source tables in the database to which any subscriber has subscribed. The user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers. Its columns are the same as those in ALL_SUBSCRIBED_TABLES.

See Also: ["ALL_SUBSCRIBED_TABLES"](#) on page 2-119

DBA_SUBSCRIPTIONS

DBA_SUBSCRIPTIONS describes all subscriptions in the database. A user must have the SELECT_CATALOG_ROLE privilege to access this view. This view is intended for use by Change Data Capture publishers. Its columns are the same as those in ALL_SUBSCRIPTIONS.

See Also: ["ALL_SUBSCRIPTIONS"](#) on page 2-120

DBA_SYNONYMS

DBA_SYNONYMS describes all synonyms in the database. Its columns are the same as those in ALL_SYNONYMS.

See Also: ["ALL_SYNONYMS"](#) on page 2-121

DBA_SYS_PRIVS

DBA_SYS_PRIVS describes system privileges granted to users and roles. This view does not display the USERNAME column.

Related View

USER_SYS_PRIVS describes system privileges granted to the current user. This view does not display the GRANTEE column, but instead displays the USERNAME column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Grantee name, user, or role receiving the grant
USERNAME	VARCHAR2(30)		Name of the current user
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege
ADMIN_OPTION	VARCHAR2(3)		Grant was with the ADMIN option

DBA_TAB_COL_STATISTICS

DBA_TAB_COL_STATISTICS contains column statistics and histogram information extracted from ["DBA_TAB_COLUMNS"](#) on page 3-129. Its columns are the same as those in ["ALL_TAB_COL_STATISTICS"](#) on page 2-121.

DBA_TAB_COLS

DBA_TAB_COLS describes the columns of all tables, views, and clusters in the database. This view differs from ["DBA_TAB_COLUMNS"](#) on page 3-129 in that hidden columns are not filtered out. Its columns are the same as those in ["ALL_TAB_COLS"](#) on page 2-122. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_TAB_COLUMNS

DBA_TAB_COLUMNS describes the columns of all tables, views, and clusters in the database. Its columns are the same as those in ["ALL_TAB_COLUMNS"](#) on page 2-124. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_TAB_COMMENTS

DBA_TAB_COMMENTS displays comments on all tables and views in the database. Its columns are the same as those in ALL_TAB_COMMENTS.

See Also: ["ALL_TAB_COMMENTS"](#) on page 2-125

DBA_TAB_HISTOGRAMS

DBA_TAB_HISTOGRAMS describes histograms on columns of all tables in the database. Its columns are the same as those in ["ALL_TAB_HISTOGRAMS"](#) on page 2-126.

DBA_TAB_MODIFICATIONS

DBA_TAB_MODIFICATIONS describes modifications to all tables in the database that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in ["ALL_TAB_MODIFICATIONS"](#) on page 2-126.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time and may not be populated until a few hours after the actual modifications occurred.

DBA_TAB_PARTITIONS

DBA_TAB_PARTITIONS provides the partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE statements for all partitions in the database. Its columns are the same as those in ["ALL_TAB_PARTITIONS"](#) on page 2-127.

DBA_TAB_PRIVS

DBA_TAB_PRIVS describes all object grants in the database.

Related View

USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object

Column	Datatype	NULL	Description
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: ["USER_TAB_PRIVS"](#) on page 3-183

DBA_TAB_STATISTICS

DBA_TAB_STATISTICS displays optimizer statistics for all tables in the database. Its columns are the same as those in ALL_TAB_STATISTICS.

See Also: ["ALL_TAB_STATISTICS"](#) on page 2-130

DBA_TAB_SUBPARTITIONS

DBA_TAB_SUBPARTITIONS describes, for each table subpartition, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in "ALL_TAB_SUBPARTITIONS" on page 2-131.

Note: Statistics are not collected on a per-subpartition basis.

DBA_TABLES

DBA_TABLES describes all relational tables in the database. Its columns are the same as those in ALL_TABLES. To gather statistics for this view, use the ANALYZE SQL statement.

See Also: ["ALL_TABLES"](#) on page 2-132

DBA_TABLESPACE_GROUPS

DBA_TABLESPACE_GROUPS describes all tablespace groups in the database.

Column	Datatype	NULL	Description
GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace group
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the temporary tablespace

DBA_TABLESPACES

DBA_TABLESPACES describes all tablespaces in the database.

Related View

USER_TABLESPACES describes the tablespaces accessible to the current user. This view does not display the PLUGGED_IN column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER	NOT NULL	Tablespace block size
INITIAL_EXTENT	NUMBER		Default initial extent size

Column	Datatype	NULL	Description
NEXT_EXTENT	NUMBER		Default incremental extent size
MIN_EXTENTS	NUMBER	NOT NULL	Default minimum number of extents
MAX_EXTENTS	NUMBER		Default maximum number of extents
PCT_INCREASE	NUMBER		Default percent increase for extent size
MIN_EXTLEN	NUMBER		Minimum extent size for this tablespace
STATUS	VARCHAR2(9)		Tablespace status: <ul style="list-style-type: none"> ▪ ONLINE ▪ OFFLINE ▪ READ ONLY
CONTENTS	VARCHAR2(9)		Tablespace contents: <ul style="list-style-type: none"> ▪ UNDO ▪ PERMANENT ▪ TEMPORARY
LOGGING	VARCHAR2(9)		Default logging attribute: <ul style="list-style-type: none"> ▪ LOGGING ▪ NOLOGGING
FORCE_LOGGING	VARCHAR2(3)		Indicates whether the tablespace is under force logging mode (YES) or not (NO)
EXTENT_MANAGEMENT	VARCHAR2(10)		Indicates whether the extents in the tablespace are dictionary managed (DICTIONARY) or locally managed (LOCAL)
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for the tablespace: <ul style="list-style-type: none"> ▪ SYSTEM ▪ UNIFORM ▪ USER
PLUGGED_IN	VARCHAR2(3)		Indicates whether the tablespace is plugged in (YES) or not (NO)
SEGMENT_SPACE_MANAGEMENT	VARCHAR2(6)		Indicates whether the free and used segment space in the tablespace is managed using free lists (MANUAL) or bitmaps (AUTO)
DEF_TAB_COMPRESSION	VARCHAR2(8)		Indicates whether default table compression is enabled (ENABLED) or not (DISABLED) Note: Enabling default table compression indicates that all tables in the tablespace will be created with table compression enabled unless otherwise specified.
RETENTION	VARCHAR2(11)		Undo tablespace retention: <ul style="list-style-type: none"> ▪ GUARANTEE - Tablespace is an undo tablespace with RETENTION specified as GUARANTEE A RETENTION value of GUARANTEE indicates that unexpired undo in all undo segments in the undo tablespace should be retained even if it means that forward going operations that need to generate undo in those segments fail. ▪ NOGUARANTEE - Tablespace is an undo tablespace with RETENTION specified as NOGUARANTEE ▪ NOT APPLY - Tablespace is not an undo tablespace
BIGFILE	VARCHAR2(3)		Indicates whether the tablespace is a bigfile tablespace (YES) or a smallfile tablespace (NO)

See Also: ["USER_TABLESPACES"](#) on page 3-184

DBA_TEMP_FILES

DBA_TEMP_FILES describes all temporary files (tempfiles) in the database.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database temp file
FILE_ID	NUMBER		File identifier number of the database temp file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file (in bytes)
BLOCKS	NUMBER		Size of the file (in Oracle blocks)
STATUS	CHAR(9)		File status: <ul style="list-style-type: none"> ▪ AVAILABLE
RELATIVE_FNO	NUMBER		Tablespace-relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Indicates whether the file is autoextensible (YES) or not (NO)
MAXBYTES	NUMBER		maximum size of the file (in bytes)
MAXBLOCKS	NUMBER		Maximum size of the file (in Oracle blocks)
INCREMENT_BY	NUMBER		Default increment for autoextension
USER_BYTES	NUMBER		Size of the useful portion of the file (in bytes)
USER_BLOCKS	NUMBER		Size of the useful portion of the file (in Oracle blocks)

DBA_THRESHOLDS

DBA_THRESHOLDS describes all thresholds.

Column	Datatype	NULL	Description
METRICS_NAME	VARCHAR2(64)		Metrics name
WARNING_OPERATOR	VARCHAR2(12)		Relational operator for warning thresholds: <ul style="list-style-type: none"> ▪ GT ▪ EQ ▪ LT ▪ LE ▪ GE ▪ CONTAINS ▪ NE ▪ DO NOT CHECK ▪ DO_NOT_CHECK
WARNING_VALUE	VARCHAR2(256)		Warning threshold value
CRITICAL_OPERATOR	VARCHAR2(12)		Relational operator for critical thresholds: <ul style="list-style-type: none"> ▪ GT ▪ EQ ▪ LT ▪ LE ▪ GE ▪ CONTAINS ▪ NE ▪ DO NOT CHECK ▪ DO_NOT_CHECK
CRITICAL_VALUE	VARCHAR2(256)		Critical threshold value

Column	Datatype	NULL	Description
OBSERVATION_PERIOD	NUMBER		Observation period length (in minutes)
CONSECUTIVE_OCCURRENCES	NUMBER		Number of occurrences before an alert is issued
INSTANCE_NAME	VARCHAR2(16)		Instance name; null for database-wide alerts
OBJECT_TYPE	VARCHAR2(64)		Object type: <ul style="list-style-type: none"> ▪ SYSTEM ▪ SERVICE ▪ EVENT_CLASS ▪ TABLESPACE ▪ FILE
OBJECT_NAME	VARCHAR2(513)		Name of the object for which the threshold is set
STATUS	VARCHAR2(7)		Indicates whether the threshold is applicable on a valid object (VALID) or not (INVALID)

DBA_TRANSFORMATIONS

DBA_TRANSFORMATIONS displays information about all transformations in the database. These transformations can be specified with Advanced Queuing operations such as enqueue, dequeue, and subscribe to automatically integrate transformations in AQ messaging.

Related View

USER_TRANSFORMATIONS displays information about transformations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TRANSFORMATION_ID	NUMBER	NOT NULL	Unique identifier for the transformation
OWNER	VARCHAR2(30)	NOT NULL	Owning user of the transformation
NAME	VARCHAR2(30)	NOT NULL	Transformation name
FROM_TYPE	VARCHAR2(61)		Source type name
TO_TYPE	VARCHAR2(91)		Target type name

See Also: ["USER_TRANSFORMATIONS"](#) on page 3-184

DBA_TRIGGER_COLS

DBA_TRIGGER_COLS lists the use of columns in all triggers in the database. Its columns are the same as those in ["ALL_TRIGGER_COLS"](#) on page 2-134.

DBA_TRIGGERS

DBA_TRIGGERS describes all triggers in the database. Its columns are the same as those in ["ALL_TRIGGERS"](#) on page 2-135.

DBA_TS_QUOTAS

DBA_TS_QUOTAS describes tablespace quotas for all users.

Related View

USER_TS_QUOTAS describes tablespace quotas for the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
USERNAME	VARCHAR2(30)	NOT NULL	User with resource rights on the tablespace
BYTES	NUMBER		Number of bytes charged to the user
MAX_BYTES	NUMBER		User's quota in bytes, or -1 if no limit
BLOCKS	NUMBER	NOT NULL	Number of Oracle blocks charged to the user
MAX_BLOCKS	NUMBER		User's quota in Oracle blocks, or -1 if no limit

DBA_TUNE_MVIEW

DBA_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure.

Related View

USER_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_NAME	VARCHAR2(30)		Name of the task
ACTION_ID	NUMBER	NOT NULL	Identifier of the action
SCRIPT_TYPE	VARCHAR2(14)		Type of the script: <ul style="list-style-type: none"> ■ IMPLEMENTATION ■ UNDO
STATEMENT	CLOB		Action statement

See Also: ["USER_TUNE_MVIEW"](#) on page 3-184

DBA_TYPE_ATTRS

DBA_TYPE_ATTRS describes the attributes of all object types in the database. Its columns are the same as those in ALL_TYPE_ATTRS.

See Also: ["ALL_TYPE_ATTRS"](#) on page 2-135

DBA_TYPE_METHODS

DBA_TYPE_METHODS describes the methods of all object types in the database. Its columns are the same as those in ALL_TYPE_METHODS.

See Also: ["ALL_TYPE_METHODS"](#) on page 2-136

DBA_TYPE_VERSIONS

DBA_TYPE_VERSIONS describes the versions of all object types in the database. Its columns are the same as those in ALL_TYPE_VERSIONS.

See Also: ["ALL_TYPE_VERSIONS"](#) on page 2-137

DBA_TYPES

DBA_TYPES describes all object types in the database. Its columns are the same as those in ALL_TYPES.

See Also: ["ALL_TYPES"](#) on page 2-137

DBA_UNDO_EXTENTS

DBA_UNDO_EXTENTS describes the extents comprising the segments in all undo tablespaces in the database.

Column	Datatype	NULL	Description
OWNER	CHAR(3)		Owner of the undo tablespace
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the undo segment
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the undo tablespace
EXTENT_ID	NUMBER		ID of the extent
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Start block number of the extent
BYTES	NUMBER		Size of the extent (in bytes)
BLOCKS	NUMBER		Size of the extent (in blocks)
RELATIVE_FNO	NUMBER		Relative number of the file containing the segment header
COMMIT_JTIME	NUMBER		Commit time of the undo in the extent expressed as Julian time
COMMIT_WTIME	VARCHAR2(20)		Commit time of the undo in the extent expressed as Wallclock time
STATUS	VARCHAR2(9)		Transaction Status of the undo in the extent: <ul style="list-style-type: none"> ▪ ACTIVE ▪ EXPIRED ▪ UNEXPIRED

DBA_UNUSED_COL_TABS

DBA_UNUSED_COL_TABS describes all tables in the database containing unused columns. Its columns are the same as those in ["ALL_UNUSED_COL_TABS"](#) on page 2-138.

DBA_UPDATABLE_COLUMNS

DBA_UPDATABLE_COLUMNS describes all columns in a join view that can be updated by the database administrator. Its columns are the same as those in ["ALL_UPDATABLE_COLUMNS"](#) on page 2-138.

See Also: *Oracle Database Concepts* for information on updatable join views

DBA_USERS

DBA_USERS describes all users of the database.

Related View

USER_USERS describes the current user. This view does not display the PASSWORD or PROFILE columns.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
PASSWORD	VARCHAR2(30)		Encrypted password
ACCOUNT_STATUS	VARCHAR2(32)	NOT NULL	Account status: <ul style="list-style-type: none"> ▪ OPEN ▪ EXPIRED ▪ EXPIRED(GRACE) ▪ LOCKED(TIMED) ▪ LOCKED ▪ EXPIRED & LOCKED(TIMED) ▪ EXPIRED(GRACE) & LOCKED(TIMED) ▪ EXPIRED & LOCKED ▪ EXPIRED(GRACE) & LOCKED
LOCK_DATE	DATE		Date the account was locked if account status was LOCKED
EXPIRY_DATE	DATE		Date of expiration of the account
DEFAULT_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for data
TEMPORARY_TABLESPACE	VARCHAR2(30)	NOT NULL	Name of the default tablespace for temporary tables or the name of a tablespace group
CREATED	DATE	NOT NULL	User creation date
PROFILE	VARCHAR2(30)	NOT NULL	User resource profile name
INITIAL_RSRC_CONSUMER_GROUP	VARCHAR2(30)		Initial resource consumer group for the user
EXTERNAL_NAME	VARCHAR2(4000)		User external name

See Also: ["USER_USERS"](#) on page 3-185

DBA_USTATS

DBA_USTATS describes all user-defined statistics collected on all objects in the database. Its columns are the same as those in ["ALL_USTATS"](#) on page 2-139.

DBA_VARRAYS

DBA_VARRAYS describes all the varrays in the database. Its columns are the same as those in ["ALL_VARRAYS"](#) on page 2-140.

DBA_VIEWS

DBA_VIEWS describes all views in the database. Its columns are the same as those in [ALL_VIEWS](#).

See Also: ["ALL_VIEWS"](#) on page 2-140

DBA_WAITERS

DBA_WAITERS shows all the sessions that are waiting for a lock, but do not hold locks for which another session is waiting.

Column	Datatype	NULL	Description
WAITING_SESSION	NUMBER		The waiting session
HOLDING_SESSION	NUMBER		The holding session
LOCK_TYPE	VARCHAR2(26)		The lock type
MODE_HELD	VARCHAR2(40)		The mode held
MODE_REQUESTED	VARCHAR2(40)		The mode requested
LOCK_ID1	VARCHAR2(40)		Lock ID 1
LOCK_ID2	VARCHAR2(40)		Lock ID 2

DBA_WARNING_SETTINGS

DBA_WARNING_SETTINGS displays information about the warning parameter settings for all objects in the database. Its columns are the same as those in ALL_WARNING_SETTINGS.

See Also: ["ALL_WARNING_SETTINGS"](#) on page 2-141

DBA_XML_SCHEMAS

DBA_XML_SCHEMAS describes all registered XML schemas in the database. Its columns are the same as those in ALL_XML_SCHEMAS.

See Also: ["ALL_XML_SCHEMAS"](#) on page 2-142

DBA_XML_TAB_COLS

DBA_XML_TAB_COLS describes the columns of all XML tables in the database. Its columns are the same as those in ALL_XML_TAB_COLS.

See Also: ["ALL_XML_TAB_COLS"](#) on page 2-142

DBA_XML_TABLES

DBA_XML_TABLES describes all XML tables in the database. Its columns are the same as those in ALL_XML_TABLES.

See Also: ["ALL_XML_TABLES"](#) on page 2-143

DBA_XML_VIEW_COLS

DBA_XML_VIEW_COLS describes the columns of all XML views in the database. Its columns are the same as those in ALL_XML_VIEW_COLS.

See Also: ["ALL_XML_VIEW_COLS"](#) on page 2-143

FLASHBACK_TRANSACTION_QUERY

FLASHBACK_TRANSACTION_QUERY displays information about all flashback transaction queries in the database.

Column	Datatype	NULL	Description
XID	RAW(8)		Transaction identifier
START_SCN	NUMBER		Transaction start system change number (SCN)
START_TIMESTAMP	DATE		Transaction start timestamp
COMMIT_SCN	NUMBER		Transaction commit system change number (null for active transactions)
COMMIT_TIMESTAMP	DATE		Transaction commit timestamp (null for active transactions)
LOGON_USER	VARCHAR2(30)		Logon user for the transaction
UNDO_CHANGE#	NUMBER		Undo system change # (1 or higher)
OPERATION	VARCHAR2(32)		Forward-going DML operation performed by the transaction: <ul style="list-style-type: none"> ▪ D - Delete ▪ I - Insert ▪ U - Update ▪ B ▪ UNKNOWN
TABLE_NAME	VARCHAR2(256)		Name of the table to which the DML applies
TABLE_OWNER	VARCHAR2(32)		Owner of the table to which the DML applies
ROW_ID	VARCHAR2(19)		Rowid of the row that was modified by the DML
UNDO_SQL	VARCHAR2(4000)		SQL to undo the DML indicated by OPERATION

GLOBAL_NAME

GLOBAL_NAME contains one row that displays the global name of the current database.

Column	Datatype	NULL	Description
GLOBAL_NAME	VARCHAR2(4000)		Global name of the database

HS_ALL_CAPS

HS_ALL_CAPS contains information about all of the capabilities (that is, features) associated with non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER		Capability number
CONTEXT	NUMBER		Context in which this capability is applicable
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Flag for internal use
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_ALL_DD

HS_ALL_DD contains data dictionary information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)		Data dictionary table name
TRANSLATION_TYPE	CHAR(1)		T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

DBA_XML_VIEWS

DBA_XML_VIEWS describes all XML views in the database. Its columns are the same as those in ALL_XML_VIEWS.

See Also: ["ALL_XML_VIEWS"](#) on page 2-144

DBMS_ALERT_INFO

DBMS_ALERT_INFO describes registered alerts.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the alert
SID	VARCHAR2(30)	NOT NULL	Session ID of a session waiting for this alert
CHANGED	VARCHAR2(1)		Boolean flag to indicate that an alert has been signaled. Y: alert signaled, N: no alert.
MESSAGE	VARCHAR2(1800)		Optional message passed by signaler

DBMS_LOCK_ALLOCATED

DBMS_LOCK_ALLOCATED describes user-allocated locks.

Column	Datatype	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	Name of the lock
LOCKID	NUMBER(38)		Lock identifier number
EXPIRATION	DATE		Planned lock expiration date (updates whenever the allocation procedure is run)

DEPTREE

This view, created by `utldtree.sql`, contains information on the object dependency tree. For user SYS, this view displays shared cursors (and only shared cursors) that depend on the object. For all other users, it displays objects other than shared cursors. Other users can access SYS.DEPTREE for information on shared cursors.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name

Column	Datatype	NULL	Description
SEQ#	NUMBER		Sequence number in the dependency tree. Used for ordering queries. See Also: "IDEPTREE" on page 3-145

DICT

DICT is a synonym for DICTIONARY.

See Also: ["DICTIONARY"](#) on page 3-140

DICT_COLUMNS

DICT_COLUMNS contains descriptions of columns in data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object that contains the column
COLUMN_NAME	VARCHAR2(30)		Name of the column
COMMENTS	VARCHAR2(4000)		Text comment on the column

DICTIONARY

DICTIONARY contains descriptions of data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object
COMMENTS	VARCHAR2(4000)		Text comment on the object

ERROR_SIZE

ERROR_SIZE is accessed to create ["DBA_OBJECT_SIZE"](#) on page 3-96 and ["USER_OBJECT_SIZE"](#) on page 3-171.

EXCEPTIONS

EXCEPTIONS contains information on violations of integrity constraints. This view is created by the `utlexcpt.sql` script.

Column	Datatype	NULL	Description
ROW_ID	ROWID		Row that caused the violation
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
CONSTRAINT	VARCHAR2(30)		Integrity constraint that was violated

HS_ALL_INITS

HS_ALL_INITS contains initialization parameter information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)		Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)		Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)		Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_BASE_CAPS

HS_BASE_CAPS contains information about base capability (that is, base features) of the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Description of the capability

HS_BASE_DD

HS_BASE_DD displays information from the base data dictionary translation table.

Column	Datatype	NULL	Description
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

HS_CLASS_CAPS

HS_CLASS_CAPS contains information about the class-specific (driver) capabilities belonging to the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Flag indicating the context in which the capability is enabled
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_CLASS_DD

HS_CLASS_DD displays information from the non-Oracle data store (FDS) class-specific data dictionary translations.

HS_CLASS_INIT

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_CLASS_INIT

HS_CLASS_INIT displays information about the non-Oracle (FDS) class-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_INIT_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_EXTERNAL_OBJECT_PRIVILEGES

HS_EXTERNAL_OBJECT_PRIVILEGES contains information about the privileges on objects that are granted to users.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object; unique for each instance
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_EXTERNAL_OBJECTS

HS_EXTERNAL_OBJECTS contains information about all of the distributed external objects accessible from the Oracle server.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
OWNER	VARCHAR(30)	NOT NULL	Name of the user who created the object

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object; unique for each instance
OBJECT_TYPE	VARCHAR2(13)	NOT NULL	Type of object: FUNCTION, PROCEDURE, PACKAGE, or LIBRARY
OBJECT_TEXT	LONG	NOT NULL	SQL text used to create the object

HS_EXTERNAL_USER_PRIVILEGES

HS_EXTERNAL_USER_PRIVILEGES contains information about all of the granted privileges that are not tied to any particular object.

Column	Datatype	NULL	Description
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_FDS_CLASS

HS_FDS_CLASS contains information about legal non-Oracle (FDS) classes.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_CLASS_COMMENTS	VARCHAR2(255)		Text description of the non-Oracle class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_FDS_INST

HS_FDS_INST contains information about non-Oracle (FDS) instances.

Column	Datatype	NULL	Description
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_COMMENTS	VARCHAR2(255)		Text description of the non-Oracle instance
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_CAPS

HS_INST_CAPS contains information about instance-specific capabilities (that is, features).

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Context in which this capability is applicable

Column	Datatype	NULL	Description
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_DD

HS_INST_DD displays information from the non-Oracle (FDS) instance-specific data dictionary translations.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_INIT

HS_INST_INIT contains information about the non-Oracle (FDS) instance-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example: ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_INIT_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

IDEPTREE

This view, created by `utldtree.sql`, lists the indented dependency tree. It is a pre-sorted, pretty-print version of `DEPTREE`.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name

IND

IND is a synonym for `USER_INDEXES`.

See Also: ["USER_INDEXES"](#) on page 3-166

INDEX_HISTOGRAM

INDEX_HISTOGRAM contains information from the `ANALYZE INDEX ... VALIDATE STRUCTURE` statement.

Note: The `ANALYZE INDEX ... VALIDATE STRUCTURE OFFLINE` statement must be used in order to collect statistics.

Column	Datatype	NULL	Description
REPEAT_COUNT	NUMBER		Number of times that one or more index keys is repeated in the table
KEYS_WITH_REPEAT_COUNT	NUMBER		Number of index keys that are repeated that many times

INDEX_STATS

INDEX_STATS stores information from the last `ANALYZE INDEX ... VALIDATE STRUCTURE` statement.

Note: The `ANALYZE INDEX ... VALIDATE STRUCTURE OFFLINE` statement must be used in order to collect statistics

Column	Datatype	NULL	Description
HEIGHT	NUMBER		Height of the B-Tree
BLOCKS	NUMBER	NOT NULL	Blocks allocated to the segment
NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition of the index which was analyzed. If the index is not partitioned, null is returned.
LF_ROWS	NUMBER		Number of leaf rows (values in the index)
LF_BLKs	NUMBER		Number of leaf blocks in the B-Tree
LF_ROWS_LEN	NUMBER		Sum of the lengths of all the leaf rows

MAP_OBJECT

Column	Datatype	NULL	Description
LF_BLK_LEN	NUMBER		Usable space in a leaf block
BR_ROWS	NUMBER		Number of branch rows in the B-Tree
BR_BLKs	NUMBER		Number of branch blocks in the B-Tree
BR_ROWS_LEN	NUMBER		Sum of the lengths of all the branch blocks in the B-Tree
BR_BLK_LEN	NUMBER		Usable space in a branch block
DEL_LF_ROWS	NUMBER		Number of deleted leaf rows in the index
DEL_LF_ROWS_LEN	NUMBER		Total length of all deleted rows in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index (may include rows that have been deleted)
MOST_REPEATED_KEY	NUMBER		How many times the most repeated key is repeated (may include rows that have been deleted)
BTREE_SPACE	NUMBER		Total space currently allocated in the B-Tree
USED_SPACE	NUMBER		Total space that is currently being used in the B-Tree
PCT_USED	NUMBER		Percent of space allocated in the B-Tree that is being used
ROWS_PER_KEY	NUMBER		Average number of rows per distinct key (this figure is calculated without consideration of deleted rows)
BLKS_GETS_PER_ACCESS	NUMBER		Expected number of consistent mode block reads per row, assuming that a randomly chosen row is accessed using the index. Used to calculate the number of consistent reads that will occur during an index scan.
PRE_ROWS	NUMBER		Number of prefix rows (values in the index)
PRE_ROWS_LEN	NUMBER		Sum of lengths of all prefix rows
OPT_CMPR_COUNT	NUMBER		Optimal key compression length
OPT_CMPR_PCTSAVE	NUMBER		Corresponding space savings after an ANALYZE

MAP_OBJECT

MAP_OBJECT displays the hierarchical arrangement of storage containers for objects. Each row in the table represents a level in the hierarchy.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(2000)		Name of the object
OBJECT_OWNER	VARCHAR2(2000)		Owner of the object
OBJECT_TYPE	VARCHAR2(2000)		Object type
FILE_MAP_IDX	NUMBER		File index (corresponds to FILE_MAP_IDX in V\$MAP_FILE)
DEPTH	NUMBER		Element depth within the I/O stack
ELEM_IDX	NUMBER		Index corresponding to the element
CU_SIZE	NUMBER		Contiguous set of logical blocks of the file (in HKB units) that is resident contiguously on the element
STRIDE	NUMBER		Number of HKB between contiguous units (CU) in the file that are contiguous on this element. Used in RAID5 and striped files.
NUM_CU	NUMBER		Number of contiguous units that are adjacent to each other on this element that are separated by STRIDE HKB in the file. In RAID5, the number of contiguous units also include the parity stripes.
ELEM_OFFSET	NUMBER		Element offset (in HKB units)

Column	Datatype	NULL	Description
FILE_OFFSET	NUMBER		Offset (in HKB units) from the start of the file to the first byte of the contiguous units
DATA_TYPE	VARCHAR2(2000)		Datatype (DATA, PARITY, or DATA AND PARITY)
PARITY_POS	NUMBER		Position of the parity. Only for RAID5. This field is needed to distinguish the parity from the data part.
PARITY_PERIOD	NUMBER		Parity period. Only for RAID5.

NLS_DATABASE_PARAMETERS

NLS_DATABASE_PARAMETERS lists permanent NLS parameters of the database.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)	NOT NULL	Parameter name
VALUE	VARCHAR2(40)		Parameter value

NLS_INSTANCE_PARAMETERS

NLS_INSTANCE_PARAMETERS lists NLS parameters of the instance.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(40)		Parameter value

NLS_SESSION_PARAMETERS

NLS_SESSION_PARAMETERS lists NLS parameters of the user session.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(40)		Parameter value

OBJ

OBJ is a synonym for USER_OBJECTS.

See Also: ["USER_OBJECTS"](#) on page 3-171

PLAN_TABLE

PLAN_TABLE is the default table for results of the EXPLAIN PLAN statement. It is created by utlxplan.sql, and it contains one row for each step in the execution plan.

Column	Datatype	NULL	Description
STATEMENT_ID	VARCHAR2(30)		Optional statement identifier specified in the EXPLAIN PLAN statement
TIMESTAMP	DATE		Date and time when the EXPLAIN PLAN statement was issued
REMARKS	VARCHAR2(80)		Place for comments that can be added to the steps of the execution plan

PLAN_TABLE

Column	Datatype	NULL	Description
OPERATION	VARCHAR2(30)		Name of the operation performed at this step
OPTIONS	VARCHAR2(255)		Options used for the operation performed at this step; the string returned by the index cost function
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_INSTANCE	NUMBER(38)		Numbered position of the object name in the original SQL statement
OBJECT_TYPE	VARCHAR2(30)		Descriptive modifier that further describes the type of object
OPTIMIZER	VARCHAR2(255)		Current mode of the optimizer
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
ID	NUMBER(38)		Identification number for this step in the execution plan
PARENT_ID	NUMBER(38)		ID of the next step that operates on the results of this step
POSITION	NUMBER(38)		Order of processing for steps with the same parent ID. For cost-based optimization, the value in the first row of the plan is the statement's execution cost. For rule-based optimization, the value is null in the first row.
COST	NUMBER(38)		Cost of the current operation estimated by the cost-based optimizer (CBO)
CARDINALITY	NUMBER(38)		Number of rows returned by the current operation (estimated by the CBO)
BYTES	NUMBER(38)		Number of bytes returned by the current operation
OTHER_TAG	VARCHAR2(255)		Describes the function of the SQL text in the OTHER column. Values for OTHER_TAG are: <ul style="list-style-type: none"> ■ SERIAL - SQL is the text of a locally-executed, serial query plan. Currently, SQL is not loaded in OTHER for this case. ■ SERIAL_FROM_REMOTE - SQL text shown in the OTHER column will be executed at a remote site ■ PARALLEL_COMBINED_WITH_PARENT - Parent of this operation is a DFO that performs both operations in the parallel execution plan ■ PARALLEL_COMBINED_WITH_CHILD - Child of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_TO_SERIAL - SQL text shown in the OTHER column is the top-level of the parallel plan. ■ PARALLEL_TO_PARALLEL - SQL text shown in the OTHER column is executed and output in parallel ■ PARALLEL_FROM_SERIAL - Operation consumes data from a serial operation and outputs it in parallel
PARTITION_START	VARCHAR2(255)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(255)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER(38)		Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Information about parallel execution slaves and parallel queries
DISTRIBUTION	VARCHAR2(30)		Distribution method
CPU_COST	NUMBER(38)		User-defined CPU cost

Column	Datatype	NULL	Description
IO_COST	NUMBER (38)		User-defined I/O cost
TEMP_SPACE	NUMBER (38)		Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2 (4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES		VARCHAR2 (4000)	Predicates used to filter rows before producing them
PROJECTION	VARCHAR2 (4000)		Expressions produced by the operation
TIME	NUMBER (38)		Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.

PLUGGABLE_SET_CHECK

PLUGGABLE_SET_CHECK contains pluggable set checks.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2 (30)		Owner of the object
OBJ1_NAME	VARCHAR2 (30)		Object 1
OBJ1_SUBNAME	VARCHAR2 (30)		SubObject1Name
OBJ1_TYPE	VARCHAR2 (15)		Object Type
TS1_NAME	VARCHAR2 (30)		Tablespace containing Object 1
OBJ2_NAME	VARCHAR2 (30)		Object Name
OBJ2_SUBNAME	VARCHAR2 (30)		SubObject2Name
OBJ2_TYPE	VARCHAR2 (15)		Object Type
OBJ2_OWNER	VARCHAR2 (30)		Object owner of second object
TS2_NAME	VARCHAR2 (30)		Tablespace containing Object 1
CONSTRAINT_NAME	VARCHAR2 (30)		Name of dependent constraint
REASON	VARCHAR2 (79)		Reason for Pluggable check violation
MESG_ID	NUMBER		The message ID

PRODUCT_COMPONENT_VERSION

PRODUCT_COMPONENT_VERSION contains version and status information for component products.

Column	Datatype	NULL	Description
PRODUCT	VARCHAR2 (64)		Product name

PROXY_USERS

Column	Datatype	NULL	Description
VERSION	VARCHAR2(64)		Version number
STATUS	VARCHAR2(64)		Status of release

PROXY_USERS

PROXY_USERS describes users who can assume the identity of other users.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of the client whose identity a proxy may assume
CLIENT	VARCHAR2(30)	NOT NULL	Name of a user which assume the identity of a client
ROLE	VARCHAR2(30)		Name of a role that a proxy user may or may not activate while acting as a client
FLAGS	VARCHAR2(35)		Indicates whether or not a proxy can activate all client roles, no client roles, or a specific client role

PSTUBTBL

This table contains information on stubs generated by the PSTUB utility so that an Oracle Forms 3.0 client can call stored procedures in the Oracle Database.

Note: The contents of this table are intended only for use by the PSTUB utility.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Schema part of the identifier of a stored procedure
DBNAME	VARCHAR2(128)		Database link part of the identifier of a stored procedure
LUN	VARCHAR2(30)		Library unit name part of the identifier of a stored procedure
LUTYPE	VARCHAR2(3)		Type of the stored procedure
LINENO	NUMBER		Line number of the stub
LINE	VARCHAR2(1800)		Text of the stub

PUBLIC_DEPENDENCY

PUBLIC_DEPENDENCY lists dependencies to and from objects, by object number.

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Object number
REFERENCED_OBJECT_ID	NUMBER	NOT NULL	Referenced object (the parent object)

PUBLICSYN

PUBLICSYN contains information on public synonyms.

Column	Datatype	NULL	Description
SNAME	VARCHAR2(30)		Name of the synonym
CREATOR	VARCHAR2(30)		Owner of the synonym

Column	Datatype	NULL	Description
TNAME	VARCHAR2 (30)		Table of which this is a synonym
DATABASE	VARCHAR2 (128)		Database in which the table resides
TABTYPE	VARCHAR2 (9)		Type of table

QUEUE_PRIVILEGES

QUEUE_PRIVILEGES shows all Advanced Queuing object privileges granted to the session.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2 (30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the object
NAME	VARCHAR2 (30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2 (30)	NOT NULL	Name of the user who performed the grant
ENQUEUE_PRIVILEGE	NUMBER		Permission to ENQUEUE to the queue
DEQUEUE_PRIVILEGE	NUMBER		Permission to DEQUEUE from the queue

RECYCLEBIN

RECYCLEBIN is a synonym for USER_RECYCLEBIN.

See Also: ["USER_RECYCLEBIN"](#) on page 3-175

RESOURCE_COST

RESOURCE_COST lists the cost for each resource.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2 (32)	NOT NULL	Name of the resource
UNIT_COST	NUMBER	NOT NULL	Cost of the resource

RESOURCE_MAP

RESOURCE_MAP describes resources. This table can be used to map resource names to resource numbers.

Column	Datatype	NULL	Description
RESOURCE#	NUMBER	NOT NULL	Numeric resource code
TYPE#	NUMBER	NOT NULL	Numeric type code
NAME	VARCHAR2 (32)	NOT NULL	Name of the resource

ROLE_ROLE_PRIVS

ROLE_ROLE_PRIVS describes the roles granted to other roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2 (30)	NOT NULL	Name of the role

Column	Datatype	NULL	Description
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Role that was granted
ADMIN_OPTION	VARCHAR2(3)		Signifies that the role was granted with ADMIN option

ROLE_SYS_PRIVS

ROLE_SYS_PRIVS describes system privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege granted to the role
ADMIN_OPTION	VARCHAR2(3)		Signifies the grant was with the ADMIN option

ROLE_TAB_PRIVS

ROLE_TAB_PRIVS describes table privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)		Name of the column, if applicable
PRIVILEGE	VARCHAR2(40)	NOT NULL	Object privilege granted to the role
GRANTABLE	VARCHAR2(3)		YES if the role was granted with ADMIN OPTION; otherwise NO

SEQ

SEQ is a synonym for USER_SEQUENCES.

See Also: ["USER_SEQUENCES"](#) on page 3-179

SESSION_CONTEXT

SESSION_CONTEXT lists attributes and the values set for the current session.

See Also: *Oracle Database SQL Reference* for information on the predefined attributes of the current session

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	The namespace that the active attribute is in
ATTRIBUTE	VARCHAR2(30)	NOT NULL	The name of the active attribute
VALUE	VARCHAR2(4000)	NOT NULL	The value of the active attribute

SESSION_PRIVS

SESSION_PRIVS lists the privileges that are currently available to the user.

Column	Datatype	NULL	Description
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the privilege

SESSION_ROLES

SESSION_ROLES lists the roles that are currently enabled to the user.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role

SOURCE_SIZE

Oracle accesses this view to create views about object size.

See Also: ["DBA_OBJECT_SIZE"](#) on page 3-96 and ["USER_OBJECT_SIZE"](#) on page 3-171

STMT_AUDIT_OPTION_MAP

STMT_AUDIT_OPTION_MAP describes auditing option type codes. This table can be used to map auditing option type numbers to type names.

Column	Datatype	NULL	Description
OPTION#	NUMBER	NOT NULL	Numeric auditing option type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of auditing option
PROPERTY	NUMBER	NOT NULL	Property flag of the auditing option

SYN

SYN is a synonym for USER_SYNONYMS.

See Also: ["ALL_SYNONYMS"](#) on page 2-121

SYNONYMS

SYNONYMS is included for compatibility. Oracle recommends that you do not use this view.

SYSCATALOG

SYSCATALOG is included for compatibility. Oracle recommends that you do not use this view.

SYSFILES

SYSFILES is included for compatibility. Oracle recommends that you do not use this view.

SYSSEGOBJ

SYSSEGOBJ is included for compatibility. Oracle recommends that you do not use this view.

SYSTEM_PRIVILEGE_MAP

SYSTEM_PRIVILEGE_MAP describes privilege type codes. This table can be used to map privilege type numbers to type names.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege
PROPERTY	NUMBER	NOT NULL	Property flag of the privilege

SYS_OBJECTS

SYS_OBJECTS maps object IDs to object types and segment data block addresses.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(15)		Type of the object
OBJECT_TYPE_ID	NUMBER		Type ID of the object
SEGMENT_TYPE_ID	NUMBER		Type of segment: TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE
OBJECT_ID	NUMBER		Object identifier
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
TS_NUMBER	NUMBER		The tablespace number

TAB

TAB is included for compatibility. Oracle recommends that you do not use this view.

TABLE_PRIVILEGE_MAP

TABLE_PRIVILEGE_MAP describes privilege (auditing option) type codes. This table can be used to map privilege (auditing option) type numbers to type names.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege (auditing option) type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege (auditing option)

TABS

TABS is a synonym for USER_TABLES.

See Also: ["USER_TABLES"](#) on page 3-184

TABQUOTAS

TABQUOTAS is included for compatibility. Oracle recommends that you do not use this view.

TRUSTED_SERVERS

TRUSTED_SERVERS displays whether a server is trusted or untrusted.

Column	Datatype	NULL	Description
TRUST	VARCHAR2(9)		Trustedness of the server listed. Values can be TRUSTED or UNTRUSTED servers which are not listed in the NAME column have opposite trustedness.
NAME	VARCHAR2(128)		Server name. Can be a specific server name or ALL for all servers.

Table 3–3 shows examples of the values returned depending on the status of the servers.

Table 3–3 TRUSTED_SERVERS Values

Condition (If . . .)	TRUSTED column	NAME column
... all servers are trusted	Trusted	ALL
... no servers are trusted	Untrusted	ALL
... all servers except DB1 are trusted	Untrusted	DB1
... all servers except DB1 are untrusted	Trusted	DB1

See Also: *Oracle Database Heterogeneous Connectivity Administrator's Guide*

TS_PITR_CHECK

This view, created by `catpitr.sql`, provides information on any dependencies or restrictions that might prevent tablespace point-in-time recovery from proceeding. This view applies only to the tablespace point-in-time recovery feature.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)	NOT NULL	The owner of the object preventing tablespace point-in-time recovery. See the REASON column for details.
OBJ1_NAME	VARCHAR2(30)	NOT NULL	The name of the object preventing tablespace point-in-time recovery
OBJ1_TYPE	VARCHAR2(15)		The object type for the object preventing tablespace point-in-time recovery
OBJ1_SUBNAME	VARCHAR2(30)		Subordinate to OBJ1_NAME
TS1_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the object preventing tablespace point-in-time recovery
OBJ2_NAME	VARCHAR2(30)		The name of a second object which may be preventing tablespace point-in-time recovery. If NULL, object 1 is the only object preventing recovery.
OBJ2_TYPE	VARCHAR2(15)		The object type for the second object (will be NULL if OBJ2_NAME is NULL)

TS_PITR_OBJECTS_TO_BE_DROPPED

Column	Datatype	NULL	Description
OBJ2_OWNER	VARCHAR2(30)		The owner of the second object (will be NULL if OBJ2_NAME is NULL)
OBJ2_SUBNAME	VARCHAR2(30)		Subordinate to OBJ2_NAME
TS2_NAME	VARCHAR2(30)		Name of the tablespace containing second object which may be preventing tablespace point-in-time recovery (-1 indicates not applicable)
CONSTRAINT_NAME	VARCHAR2(30)		Name of the constraint
REASON	VARCHAR2(78)		Reason why tablespace point-in-time recovery cannot proceed

See Also: *Oracle Database Backup and Recovery Basics*

TS_PITR_OBJECTS_TO_BE_DROPPED

TS_PITR_OBJECTS_TO_BE_DROPPED lists all objects lost as a result of performing tablespace point-in-time recovery. This view applies only to the tablespace point-in-time recovery feature.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
NAME	VARCHAR2(30)	NOT NULL	The name of the object that will be lost as a result of undergoing tablespace point-in-time recovery
CREATION_TIME	DATE	NOT NULL	Creation timestamp of the object
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the object

UNI_PLUGGABLE_SET_CHECK

UNI_PLUGGABLE_SET_CHECK contains pluggable check information.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name
OBJ1_TYPE	VARCHAR2(15)		Object Type
TS1_NAME	VARCHAR2(30)		Tablespace containing Object 1
OBJ2_NAME	VARCHAR2(30)		Object Name
OBJ2_SUBNAME	VARCHAR2(30)		SubObject2Name
OBJ2_TYPE	VARCHAR2(15)		Object Type
OBJ2_OWNER	VARCHAR2(30)		Object owner of second object
TS2_NAME	VARCHAR2(30)		Tablespace containing Object 1
CONSTRAINT_NAME	VARCHAR2(30)		Name of dependent constraint
REASON	VARCHAR2(79)		Reason for Pluggable check violation
MSG_ID	NUMBER		The message ID

USER_ADVISOR_ACTIONS

USER_ADVISOR_ACTIONS displays information about the actions associated with the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_ACTIONS.

See Also: ["DBA_ADVISOR_ACTIONS"](#) on page 3-2

USER_ADVISOR_FINDINGS

USER_ADVISOR_FINDINGS displays the findings discovered by the advisors owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_FINDINGS.

See Also: ["DBA_ADVISOR_FINDINGS"](#) on page 3-4

USER_ADVISOR_JOURNAL

USER_ADVISOR_JOURNAL displays the journal entries for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_JOURNAL.

See Also: ["DBA_ADVISOR_JOURNAL"](#) on page 3-4

USER_ADVISOR_LOG

USER_ADVISOR_LOG displays information about the current state of the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_LOG.

See Also: ["DBA_ADVISOR_LOG"](#) on page 3-5

USER_ADVISOR_OBJECTS

USER_ADVISOR_OBJECTS displays information about the objects currently referenced by the advisors owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_OBJECTS.

See Also: ["DBA_ADVISOR_OBJECTS"](#) on page 3-6

USER_ADVISOR_PARAMETERS

USER_ADVISOR_PARAMETERS displays the parameters and their current values for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_PARAMETERS.

See Also: ["DBA_ADVISOR_PARAMETERS"](#) on page 3-7

USER_ADVISOR_RATIONALE

USER_ADVISOR_RATIONALE displays information about the rationales for the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_RATIONALE.

See Also: ["DBA_ADVISOR_RATIONALE"](#) on page 3-8

USER_ADVISOR_RECOMMENDATIONS

USER_ADVISOR_RECOMMENDATIONS displays the results of an analysis of the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_RECOMMENDATIONS.

See Also: ["DBA_ADVISOR_RECOMMENDATIONS"](#) on page 3-9

USER_ADVISOR_SQLA_REC_SUM

USER_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_REC_SUM.

See Also: ["DBA_ADVISOR_SQLA_REC_SUM"](#) on page 3-10

USER_ADVISOR_SQLA_WK_MAP

USER_ADVISOR_SQLA_WK_MAP displays the workload references for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_WK_MAP.

See Also: ["DBA_ADVISOR_SQLA_WK_MAP"](#) on page 3-10

USER_ADVISOR_SQLA_WK_STMTS

USER_ADVISOR_SQLA_WK_STMTS displays information about the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_WK_STMTS.

See Also: ["DBA_ADVISOR_SQLA_WK_STMTS"](#) on page 3-11

USER_ADVISOR_SQLW_JOURNAL

USER_ADVISOR_SQLW_JOURNAL displays the journal entries for the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_JOURNAL.

See Also: ["DBA_ADVISOR_SQLW_JOURNAL"](#) on page 3-12

USER_ADVISOR_SQLW_PARAMETERS

USER_ADVISOR_SQLW_PARAMETERS displays the workload parameters and their current values owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_PARAMETERS.

See Also: ["DBA_ADVISOR_SQLW_PARAMETERS"](#) on page 3-12

USER_ADVISOR_SQLW_STMTS

USER_ADVISOR_SQLW_STMTS displays rows that correspond to the statements in the workload owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_STMTS.

See Also: ["DBA_ADVISOR_SQLW_STMTS"](#) on page 3-13

USER_ADVISOR_SQLW_SUM

USER_ADVISOR_SQLW_SUM displays an aggregated picture of the SQLWkld workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_SUM.

See Also: ["DBA_ADVISOR_SQLW_SUM"](#) on page 3-14

USER_ADVISOR_SQLW_TABLES

USER_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_TABLES.

See Also: ["DBA_ADVISOR_SQLW_TABLES"](#) on page 3-15

USER_ADVISOR_SQLW_TEMPLATES

USER_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of the SQLWkld template objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_TEMPLATES.

See Also: ["DBA_ADVISOR_SQLW_TEMPLATES"](#) on page 3-15

USER_ADVISOR_TASKS

USER_ADVISOR_TASKS displays information about the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_TASKS.

See Also: ["DBA_ADVISOR_TASKS"](#) on page 3-16

USER_ADVISOR_TEMPLATES

USER_ADVISOR_TEMPLATES displays information about the templates owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_TEMPLATES.

See Also: ["DBA_ADVISOR_TEMPLATES"](#) on page 3-17

USER_ALL_TABLES

USER_ALL_TABLES describes the object tables and relational tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ALL_TABLES.

See Also: ["ALL_ALL_TABLES"](#) on page 2-4

USER_AQ_AGENT_PRIVS

USER_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to the current user. Its columns (except for DB_USERNAME) are the same as those in DBA_AQ_AGENT_PRIVS.

See Also: ["DBA_AQ_AGENT_PRIVS"](#) on page 3-21

USER_ARGUMENTS

USER_ARGUMENTS lists the arguments of the procedures and functions that are owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ARGUMENTS.

See Also: ["ALL_ARGUMENTS"](#) on page 2-12

USER_ASSOCIATIONS

USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user. Its columns are the same as those in "ALL_ASSOCIATIONS" on page 2-13.

USER_ATTRIBUTE_TRANSFORMATIONS

USER_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for the transformations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ATTRIBUTE_TRANSFORMATIONS.

See Also: ["DBA_ATTRIBUTE_TRANSFORMATIONS"](#) on page 3-22

USER_AUDIT_OBJECT

This view, created by `cataudit.sql`, lists audit trail records for statements concerning objects that are accessible to the current user. Its columns are the same as those in "DBA_AUDIT_OBJECT" on page 3-23.

USER_AUDIT_POLICIES

USER_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views owned by the current user. Its columns (except for OBJECT_SCHEMA) are the same as those in ALL_AUDIT_POLICIES.

See Also: ["ALL_AUDIT_POLICIES"](#) on page 2-14

USER_AUDIT_POLICY_COLUMNS

USER_AUDIT_POLICY_COLUMNS describes the fine-grained auditing policy columns on the tables and views owned by the current user. Its columns (except for OBJECT_SCHEMA) are the same as those in ALL_AUDIT_POLICY_COLUMNS.

See Also: ["ALL_AUDIT_POLICY_COLUMNS"](#) on page 2-14

USER_AUDIT_SESSION

This view, created by `cataudit.sql`, lists all audit trail records concerning connections and disconnections for the user. Its columns are the same as those in "DBA_AUDIT_SESSION" on page 3-25.

USER_AUDIT_STATEMENT

This view, created by `cataudit.sql`, lists audit trail entries for the following statements issued by the user: GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM. Its columns are the same as those in ["DBA_AUDIT_STATEMENT"](#) on page 3-26.

USER_AUDIT_TRAIL

This view, created by `cataudit.sql`, lists audit trail entries relevant to the user. Its columns are the same as those in ["DBA_AUDIT_TRAIL"](#) on page 3-27.

USER_AW_PS

USER_AW_PS describes the pagespaces in the analytic workspaces owned by the current user. Its columns (except for OWNER) are the same as those in ALL_AW_PS.

See Also: ["ALL_AW_PS"](#) on page 2-15

USER_AWS

USER_AWS describes the analytic workspaces owned by the current user. Its columns (except for OWNER) are the same as those in ALL_AWS.

See Also: ["ALL_AWS"](#) on page 2-15

USER_BASE_TABLE_MVIEWS

USER_BASE_TABLE_MVIEWS describes the materialized views using materialized view logs owned by the current user. Its columns are the same as those in ALL_BASE_TABLE_MVIEWS.

See Also: ["ALL_BASE_TABLE_MVIEWS"](#) on page 2-16

USER_CATALOG

USER_CATALOG lists indexes, tables, views, clusters, synonyms, and sequences owned by the current user. Its columns are the same as those in ["ALL_CATALOG"](#) on page 2-20.

USER_CLU_COLUMNS

USER_CLU_COLUMNS maps columns in the current user's tables to cluster columns. Its columns are the same as those in ["DBA_CLU_COLUMNS"](#) on page 3-31.

USER_CLUSTER_HASH_EXPRESSIONS

USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for the hash clusters owned by the current user. Its columns are the same as those in ["ALL_CLUSTER_HASH_EXPRESSIONS"](#) on page 2-20.

USER_CLUSTERS

USER_CLUSTERS describes all the clusters owned by the current user. Its columns are the same as those in ["ALL_CLUSTERS"](#) on page 2-20.

USER_COL_COMMENTS

USER_COL_COMMENTS displays comments on the columns of the tables and views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_COL_COMMENTS.

See Also: ["ALL_COL_COMMENTS"](#) on page 2-21

USER_COL_PRIVS

USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee. Its columns are the same as those in DBA_COL_PRIVS.

See Also: ["DBA_COL_PRIVS"](#) on page 3-31

USER_COL_PRIVS_MADE

USER_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner. Its columns (except for OWNER) are the same as those in ALL_COL_PRIVS_MADE.

See Also: ["ALL_COL_PRIVS_MADE"](#) on page 2-22

USER_COL_PRIVS_RECD

USER_COL_PRIVS_RECD describes the column object grants for which the current user is the grantee. Its columns (except for GRANTEE) are the same as those in ALL_COL_PRIVS_RECD.

See Also: ["ALL_COL_PRIVS_RECD"](#) on page 2-23

USER_COLL_TYPES

USER_COLL_TYPES describes named collection types (VARRAYs, nested tables, object tables, and so on) in the current user's schema. Its columns are the same as those in ["ALL_COLL_TYPES"](#) on page 2-23.

USER_CONS_COLUMNS

USER_CONS_COLUMNS describes columns that are owned by the current user and that are specified in constraint definitions. Its columns are the same as those in ["ALL_CONS_COLUMNS"](#) on page 2-24.

USER_CONS_OBJ_COLUMNS

USER_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CONS_OBJ_COLUMNS.

See Also: ["ALL_CONS_OBJ_COLUMNS"](#) on page 2-24

USER_CONSTRAINTS

USER_CONSTRAINTS describes all constraint definitions on tables owned by the current user. Its columns are the same as those in ["ALL_CONSTRAINTS"](#) on page 2-25.

USER_DATAPUMP_JOBS

USER_DATAPUMP_JOBS displays the Data Pump jobs owned by the current user. Its columns (except for OWNER_NAME) are the same as those in DBA_DATAPUMP_JOBS.

See Also: ["DBA_DATAPUMP_JOBS"](#) on page 3-34

USER_DB_LINKS

USER_DB_LINKS describes the database links owned by the current user. Its columns (except for OWNER) are the same as those in ALL_DB_LINKS.

See Also: ["ALL_DB_LINKS"](#) on page 2-26

USER_DEPENDENCIES

USER_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers owned by the current user, including dependencies on views created without any database links. Its columns are the same as those in ["ALL_DEPENDENCIES"](#) on page 2-28.

USER_DIM_ATTRIBUTES

USER_DIM_ATTRIBUTES describes the relationship between dimension levels and functionally dependent columns in the current user's schema. The level columns and the dependent column must be in the same table. This view's columns are the same as those in ["ALL_DIM_ATTRIBUTES"](#) on page 2-28.

USER_DIM_CHILD_OF

USER_DIM_CHILD_OF describes a hierarchical relationship of 1 to n between pairs of levels in dimensions owned by the current user. Its columns are the same as those in ["ALL_DIM_CHILD_OF"](#) on page 2-29.

USER_DIM_HIERARCHIES

USER_DIM_HIERARCHIES describes the dimension hierarchies owned by the current user. Its columns are the same as those in ["ALL_DIM_HIERARCHIES"](#) on page 2-29.

USER_DIM_JOIN_KEY

USER_DIM_JOIN_KEY describes the join between two dimension tables owned by the current user. The join is always specified between a parent dimension level column and a child column. This view's columns are the same as those in ["ALL_DIM_JOIN_KEY"](#) on page 2-29.

USER_DIM_LEVEL_KEY

USER_DIM_LEVEL_KEY describes columns of dimension levels owned by the current user. This view's columns are the same as those in "[ALL_DIM_LEVEL_KEY](#)" on page 2-30.

USER_DIM_LEVELS

USER_DIM_LEVELS describes the levels of dimensions owned by the current user. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "[ALL_DIM_LEVELS](#)" on page 2-30.

USER_DIMENSIONS

USER_DIMENSIONS describes dimension objects in the user's schema. Its columns are the same as those in "[ALL_DIMENSIONS](#)" on page 2-31.

USER_ERRORS

USER_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) owned by the current user. Its columns are the same as those in "[ALL_ERRORS](#)" on page 2-31.

USER_EVALUATION_CONTEXT_TABLES

USER_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in [ALL_EVALUATION_CONTEXT_TABLES](#).

See Also: "[ALL_EVALUATION_CONTEXT_TABLES](#)" on page 2-32

USER_EVALUATION_CONTEXT_VARS

USER_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in [ALL_EVALUATION_CONTEXT_VARS](#).

See Also: "[ALL_EVALUATION_CONTEXT_VARS](#)" on page 2-33

USER_EVALUATION_CONTEXTS

USER_EVALUATION_CONTEXTS describes the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in [ALL_EVALUATION_CONTEXTS](#).

See Also: "[ALL_EVALUATION_CONTEXTS](#)" on page 2-33

USER_EXTENTS

USER_EXTENTS describes the extents comprising the segments owned by the current user's objects. Its columns (except for OWNER, FILE_ID, BLOCK_ID, and RELATIVE_FNO) are the same as those in [DBA_EXTENTS](#).

See Also: "[DBA_EXTENTS](#)" on page 3-40

USER_EXTERNAL_LOCATIONS

USER_EXTERNAL_TABLES describes the locations (data sources) of the external tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_EXTERNAL_LOCATIONS.

See Also: "[ALL_EXTERNAL_LOCATIONS](#)" on page 2-34

USER_EXTERNAL_TABLES

USER_EXTERNAL_TABLES describes the external tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_EXTERNAL_TABLES.

See Also: "[ALL_EXTERNAL_TABLES](#)" on page 2-34

USER_FREE_SPACE

USER_FREE_SPACE describes the free extents in the tablespaces accessible to the current user. Its columns are the same as those in DBA_FREE_SPACE.

See Also: "[DBA_FREE_SPACE](#)" on page 3-44

USER_IND_COLUMNS

USER_IND_COLUMNS describes the columns of the indexes owned by the current user and columns of indexes on tables owned by the current user. Its columns are the same as those in "[ALL_IND_COLUMNS](#)" on page 2-35.

USER_IND_EXPRESSIONS

USER_IND_EXPRESSIONS describes expressions of function-based indexes on tables owned by the current user. Its columns are the same as those in "[ALL_IND_EXPRESSIONS](#)" on page 2-36.

USER_IND_PARTITIONS

USER_IND_PARTITIONS describes, for each index partition owned by the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics analyzed by ANALYZE statements. Its columns are the same as those in "[ALL_IND_PARTITIONS](#)" on page 2-36.

USER_IND_STATISTICS

USER_IND_STATISTICS displays optimizer statistics for the indexes on the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_IND_STATS.

See Also: "[ALL_IND_STATISTICS](#)" on page 2-37

USER_IND_SUBPARTITIONS

USER_IND_SUBPARTITIONS describes, for each index subpartition owned by the current user, the partition-level partitioning information, the storage parameters for

the subpartition, and various partition statistics collected by `ANALYZE` statements. Its columns are the same as those in "[ALL_IND_SUBPARTITIONS](#)" on page 2-39.

USER_INDEXES

`USER_INDEXES` describes indexes owned by the current user. To gather statistics for this view, use the SQL `ANALYZE` statement. This view supports parallel partitioned index scans. Its columns (except for `OWNER`) are the same as those in "[ALL_INDEXES](#)" on page 2-40.

USER_INDEXTYPE_ARRAYTYPES

`USER_INDEXTYPE_ARRAYTYPES` displays information about the array types specified by the indextypes owned by the current user. Its columns are the same as those in `ALL_INDEXTYPE_ARRAYTYPES`.

See Also: "[ALL_INDEXTYPE_ARRAYTYPES](#)" on page 2-43

USER_INDEXTYPE_COMMENTS

`USER_INDEXTYPE_COMMENTS` displays comments for the user-defined indextypes owned by the current user. Its columns are the same as those in `ALL_INDEXTYPE_COMMENTS`.

See Also: "[ALL_INDEXTYPE_COMMENTS](#)" on page 2-43

USER_INDEXTYPE_OPERATORS

`USER_INDEXTYPE_OPERATORS` lists all the operators supported by indextypes owned by the current user. Its columns are the same as those in `ALL_INDEXTYPE_OPERATORS`.

See Also: "[ALL_INDEXTYPE_OPERATORS](#)" on page 2-44

USER_INDEXTYPES

`USER_INDEXTYPES` describes the indextypes owned by the current user. Its columns are the same as those in `ALL_INDEXTYPES`.

See Also: "[ALL_INDEXTYPES](#)" on page 2-44

USER_INTERNAL_TRIGGERS

`USER_INTERNAL_TRIGGERS` describes the internal triggers on all tables owned by the current user. Its columns are the same as those in "[ALL_INTERNAL_TRIGGERS](#)" on page 2-45.

USER_JAVA_ARGUMENTS

`USER_JAVA_ARGUMENTS` displays argument information about the stored Java classes owned by the current user. Its columns (except for `OWNER`) are the same as those in `ALL_JAVA_ARGUMENTS`.

See Also: "[ALL_JAVA_ARGUMENTS](#)" on page 2-45

USER_JAVA_CLASSES

USER_JAVA_CLASSES displays class level information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_CLASSES.

See Also: ["ALL_JAVA_CLASSES"](#) on page 2-46

USER_JAVA_DERIVATIONS

USER_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_DERIVATIONS.

See Also: ["ALL_JAVA_DERIVATIONS"](#) on page 2-47

USER_JAVA_FIELDS

USER_JAVA_FIELDS displays field information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_FIELDS.

See Also: ["ALL_JAVA_FIELDS"](#) on page 2-47

USER_JAVA_IMPLEMENTES

USER_JAVA_IMPLEMENTES describes interfaces implemented by the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_IMPLEMENTES.

See Also: ["ALL_JAVA_IMPLEMENTES"](#) on page 2-48

USER_JAVA_INNERS

USER_JAVA_INNERS displays information about inner classes referred to by the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_INNERS.

See Also: ["ALL_JAVA_INNERS"](#) on page 2-49

USER_JAVA_LAYOUTS

USER_JAVA_LAYOUTS displays class layout information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_LAYOUTS.

See Also: ["ALL_JAVA_LAYOUTS"](#) on page 2-49

USER_JAVA_METHODS

USER_JAVA_METHODS displays method information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_METHODS.

See Also: ["ALL_JAVA_METHODS"](#) on page 2-50

USER_JAVA_NCOMPS

USER_JAVA_NCOMPS displays ncomp-related information about the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_NCOMPS.

See Also: ["ALL_JAVA_NCOMPS"](#) on page 2-51

USER_JAVA_POLICY

USER_JAVA_POLICY describes Java security permissions for the current user. Its columns are the same as those in DBA_JAVA_POLICY.

See Also: ["DBA_JAVA_POLICY"](#) on page 3-84

USER_JAVA_RESOLVERS

USER_JAVA_RESOLVERS displays information about resolvers of the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_RESOLVERS.

See Also: ["ALL_JAVA_RESOLVERS"](#) on page 2-52

USER_JAVA_THROWS

USER_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_THROWS.

See Also: ["ALL_JAVA_THROWS"](#) on page 2-52

USER_JOBS

USER_JOBS describes all jobs owned by the user. Its columns are the same as those in ["ALL_JOBS"](#) on page 2-53.

See Also: *Oracle Database Administrator's Guide* for more information on jobs

USER_JOIN_IND_COLUMNS

USER_JOIN_IND_COLUMNS describes all join conditions owned by the current user. Its columns are the same as those in ["ALL_JOIN_IND_COLUMNS"](#) on page 2-54.

USER_LIBRARIES

USER_LIBRARIES describes all libraries owned by the current user. Its columns are the same as those in ["ALL_LIBRARIES"](#) on page 2-54.

USER_LOB_PARTITIONS

USER_LOB_PARTITIONS displays the LOB partitions contained in tables owned by the current user. Its columns are the same as those in ["ALL_LOB_PARTITIONS"](#) on page 2-54.

USER_LOB_SUBPARTITIONS

USER_LOB_SUBPARTITIONS describes partition-level attributes of LOB data subpartitions owned by the current user. Its columns are the same as those in "[ALL_LOB_SUBPARTITIONS](#)" on page 2-55.

- DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.

USER_LOB_TEMPLATES

USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. Its columns (except for USER_NAME) are the same as those in ALL_LOB_TEMPLATES.

See Also: "[ALL_LOB_TEMPLATES](#)" on page 2-56

USER_LOBS

USER_LOBS displays the user's CLOBs and BLOBs contained in the user's tables. BFILEs are stored outside the database, so they are not described by this view. This view's columns are the same as those in "[ALL_LOBS](#)" on page 2-57.

USER_LOG_GROUP_COLUMNS

USER_LOG_GROUP_COLUMNS describes columns that are owned by the current user and that are specified in log groups. Its columns are the same as those in "[ALL_LOG_GROUP_COLUMNS](#)" on page 2-58.

USER_LOG_GROUPS

USER_LOG_GROUPS describes log group definitions on tables owned by the current user. Its columns are the same as those in "[ALL_LOG_GROUPS](#)" on page 2-58.

USER_METHOD_PARAMS

USER_METHOD_PARAMS describes the method parameters of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_METHOD_PARAMS.

See Also: "[ALL_METHOD_PARAMS](#)" on page 2-59

USER_METHOD_RESULTS

USER_METHOD_RESULTS describes the method results of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_METHOD_RESULTS.

See Also: "[ALL_METHOD_RESULTS](#)" on page 2-59

USER_MVIEW_AGGREGATES

USER_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of aggregated materialized views owned by the current user. Its columns are the same as those in "[ALL_MVIEW_AGGREGATES](#)" on page 2-60.

USER_MVIEW_ANALYSIS

USER_MVIEW_ANALYSIS describes all materialized views owned by the current user that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in "[ALL_MVIEW_ANALYSIS](#)" on page 2-61.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as SYSDATE or USER. This view also excludes materialized views that were created as snapshots prior to Oracle8i and that were never altered to enable query rewrite.

USER_MVIEW_COMMENTS

USER_MVIEW_COMMENTS displays comments on the materialized views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MVIEW_COMMENTS.

See Also: "[ALL_MVIEW_COMMENTS](#)" on page 2-62

USER_MVIEW_DETAIL_RELATIONS

USER_MVIEW_DETAIL_RELATIONS represents the named detail relations that are either in the FROM list of a materialized view, or that are indirectly referenced through views in the FROM list. Its columns are the same as those in "[ALL_MVIEW_DETAIL_RELATIONS](#)" on page 2-62.

USER_MVIEW_JOINS

USER_MVIEW_JOINS describes a join between two columns in the WHERE clause of a subquery that defines a materialized view. Its columns are the same as those in "[ALL_MVIEW_JOINS](#)" on page 2-63.

USER_MVIEW_KEYS

USER_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views in the current user's schema are based. Its columns are the same as those in "[ALL_MVIEW_KEYS](#)" on page 2-63.

USER_MVIEW_LOGS

USER_MVIEW_LOGS describes all materialized view logs owned by the current user. Its columns are the same as those in ALL_MVIEW_LOGS.

See Also: "[ALL_MVIEW_LOGS](#)" on page 2-64

USER_MVIEW_REFRESH_TIMES

USER_MVIEW_REFRESH_TIMES describes refresh times of the materialized views owned by the current user. Its columns are the same as those in ALL_MVIEW_REFRESH_TIMES.

See Also: ["ALL_MVIEW_REFRESH_TIMES"](#) on page 2-65

USER_MVIEWS

USER_MVIEWS describes all materialized views owned by the current user. Its columns are the same as those in ALL_MVIEWS.

See Also: ["ALL_MVIEWS"](#) on page 2-65

USER_NESTED_TABLE_COLS

USER_NESTED_TABLE_COLS describes the columns of the nested tables owned by the current user. Its columns (except for OWNER) are the same as those in ["ALL_NESTED_TABLE_COLS"](#) on page 2-68. To gather statistics for this view, use the ANALYZE SQL statement.

USER_NESTED_TABLES

USER_NESTED_TABLES describes the nested tables in tables owned by the current user. Its columns are the same as those in ["ALL_NESTED_TABLES"](#) on page 2-70.

USER_OBJ_AUDIT_OPTS

This view, created by `cataudit.sql`, lists auditing characteristics for all objects owned by the current user. Its columns are the same as those in ["DBA_OBJ_AUDIT_OPTS"](#) on page 3-95.

USER_OBJ_COLATTRS

USER_OBJ_COLATTRS describes object columns and attributes contained in the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_OBJ_COLATTRS.

See Also: ["ALL_OBJ_COLATTRS"](#) on page 2-70

USER_OBJECT_SIZE

USER_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects. Its columns are the same as those in ["DBA_OBJECT_SIZE"](#) on page 3-96.

USER_OBJECT_TABLES

USER_OBJECT_TABLES describes the object tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_OBJECT_TABLES.

See Also: ["ALL_OBJECT_TABLES"](#) on page 2-71

USER_OBJECTS

USER_OBJECTS describes all objects owned by the current user. Its columns are the same as those in ["ALL_OBJECTS"](#) on page 2-73.

USER_OPANCILLARY

USER_OPANCILLARY provides ancillary information for operators owned by the current user. Its columns are the same as those in "ALL_OPANCILLARY" on page 2-74.

USER_OPARGUMENTS

USER_OPARGUMENTS provides argument information for operator bindings owned by the current user. Its columns are the same as those in "ALL_OPARGUMENTS" on page 2-74.

USER_OPBINDINGS

USER_OPBINDINGS describes the binding functions and methods on the operators owned by the current user. Its columns are the same as those in ALL_OPBINDINGS.

See Also: "ALL_OPBINDINGS" on page 2-75

USER_OPERATOR_COMMENTS

USER_OPERATOR_COMMENTS displays comments for the user-defined operators owned by the current user. Its columns are the same as those in ALL_INDEXTYPE_COMMENTS.

See Also: "ALL_OPERATOR_COMMENTS" on page 2-75

USER_OPERATORS

USER_OPERATORS describes all operators owned by the current user. Its columns are the same as those in ALL_OPERATORS.

See Also: "ALL_OPERATORS" on page 2-76

USER_OUTLINE_HINTS

USER_OUTLINE_HINTS describes the set of hints that make up the outlines owned by the current user. Its columns are the same as those in "ALL_OUTLINE_HINTS" on page 2-76.

USER_OUTLINES

USER_OUTLINES describes all outlines owned by the current user. Its columns are the same as those in "ALL_OUTLINES" on page 2-77.

USER_PART_COL_STATISTICS

USER_PART_COL_STATISTICS provides column statistics and histogram information for table partitions owned by the current user. Its columns are the same as those in "ALL_PART_COL_STATISTICS" on page 2-77.

USER_PART_HISTOGRAMS

USER_PART_HISTOGRAMS contains the histogram data (end-points per histogram) for histograms on table partitions that the current user can access. Its columns are the same as those in ["ALL_PART_HISTOGRAMS"](#) on page 2-78.

USER_PART_INDEXES

USER_PART_INDEXES displays the object-level partitioning information for the partitioned indexes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_INDEXES.

See Also: ["ALL_PART_INDEXES"](#) on page 2-79

USER_PART_KEY_COLUMNS

USER_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_KEY_COLUMNS.

See Also: ["ALL_PART_KEY_COLUMNS"](#) on page 2-80

USER_PART_LOBS

USER_PART_LOBS provides table-level information for partitioned LOBs owned by the current user, including default attributes for LOB data partitions. Its columns are the same as those in ["ALL_PART_LOBS"](#) on page 2-81.

USER_PART_TABLES

USER_PART_TABLES displays the object-level partitioning information for the partitioned tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_TABLES.

See Also: ["ALL_PART_TABLES"](#) on page 2-82

USER_PARTIAL_DROP_TABS

USER_PARTIAL_DROP_TABS describes all tables in the schema of the current user that have partially completed DROP COLUMN operations. Its columns are the same as those in ["ALL_PARTIAL_DROP_TABS"](#) on page 2-83.

USER_PASSWORD_LIMITS

USER_PASSWORD_LIMITS describes the password profile parameters that are assigned to the user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the password resource
LIMIT	VARCHAR2(40)		Value of the resource limit

USER_PENDING_CONV_TABLES

USER_PENDING_CONV_TABLES describes the pending conversion tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PENDING_CONV_TABLES.

See Also: ["ALL_PENDING_CONV_TABLES"](#) on page 2-83

USER_PLSQL_OBJECT_SETTINGS

USER_PLSQL_OBJECT_SETTINGS displays compiler settings for the stored objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PLSQL_OBJECT_SETTINGS.

See Also: ["ALL_PLSQL_OBJECT_SETTINGS"](#) on page 2-84

USER_POLICIES

USER_POLICIES describes the security policies on the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICIES.

See Also: ["ALL_POLICIES"](#) on page 2-85

USER_POLICY_CONTEXTS

USER_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICY_CONTEXTS.

See Also: ["ALL_POLICY_CONTEXTS"](#) on page 2-86

USER_POLICY_GROUPS

USER_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICY_GROUPS.

See Also: ["ALL_POLICY_GROUPS"](#) on page 2-86

USER_PROCEDURES

USER_PROCEDURES lists all functions and procedures, along with their associated properties. Its columns (except OWNER) are the same as those in ["ALL_PROCEDURES"](#) on page 2-87.

USER_PROXIES

USER_PROXIES displays information about connections the current user is allowed to proxy. Its columns (except for PROXY) are the same as those in DBA_PROXIES.

See Also: ["DBA_PROXIES"](#) on page 3-101

USER_PUBLISHED_COLUMNS

USER_PUBLISHED_COLUMNS describes the published source table columns owned by the current user. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in ALL_PUBLISHED_COLUMNS.

See Also: ["ALL_PUBLISHED_COLUMNS"](#) on page 2-88

USER_QUEUE_SCHEDULES

USER_QUEUE_SCHEDULES lists information about queue schedules. Its columns are the same as those in "DBA_QUEUE_SCHEDULES" on page 3-102.

USER_QUEUE_TABLES

USER_QUEUE_TABLES describes the queues in the queue tables created in the current user's schema. Its columns (except for OWNER) are the same as those in ALL_QUEUE_TABLES.

See Also: ["ALL_QUEUE_TABLES"](#) on page 2-88

USER_QUEUES

USER_QUEUES describes the operational characteristics of every queue in the user's schema. Its columns are the same as those in "ALL_QUEUES" on page 2-89.

See Also: *Oracle Streams Advanced Queuing User's Guide and Reference* for more information about these views and Advanced Queuing

USER_RECYCLEBIN

USER_RECYCLEBIN displays information about the recycle bin owned by the current user. Its columns (except for OWNER) are the same as those in DBA_RECYCLEBIN.

See Also: ["DBA_RECYCLEBIN"](#) on page 3-104

USER_REFRESH

USER_REFRESH describes all refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH" on page 2-90.

USER_REFRESH_CHILDREN

USER_REFRESH_CHILDREN lists all the objects in refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 2-91.

USER_REFS

USER_REFS describes the REF columns and REF attributes in the object type columns of tables owned by the current user. Its columns are the same as those in "ALL_REFS" on page 2-92.

USER_REGISTERED_MVIEWS

USER_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) owned by the current user. Its columns are the same as those in ALL_REGISTERED_MVIEWS.

See Also: ["ALL_REGISTERED_MVIEWS"](#) on page 2-92

USER_REGISTRY

USER_REGISTRY displays information about the components loaded into the database that are owned by the current user. Its columns are the same as those in DBA_REGISTRY.

See Also: ["DBA_REGISTRY"](#) on page 3-107

USER_RESOURCE_LIMITS

USER_RESOURCE_LIMITS displays the resource limits for the current user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2 (32)	NOT NULL	Name of the resource
LIMIT	VARCHAR2 (40)		Limit placed on this resource

USER_RESUMABLE

USER_RESUMABLE lists resumable statements executed by the current user. Its columns are the same as those in DBA_RESUMABLE.

See Also: ["DBA_RESUMABLE"](#) on page 3-109

USER_REWRITE_EQUIVALENCES

USER_REWRITE_EQUIVALENCES describes the rewrite equivalences owned by the current user. Its columns are the same as those in ALL_REWRITE_EQUIVALENCES.

See Also: ["ALL_REWRITE_EQUIVALENCES"](#) on page 2-93

USER_ROLE_PRIVS

USER_ROLE_PRIVS describes the roles granted to the current user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2 (30)		Name of the user, or PUBLIC
GRANTED_ROLE	VARCHAR2 (30)		Name of the role granted to the user
ADMIN_OPTION	VARCHAR2 (3)		Indicates whether the grant was with the ADMIN OPTION (YES) or not (NO)
DEFAULT_ROLE	VARCHAR2 (3)		Indicates whether the role is designated as a DEFAULT ROLE for the user (YES) or not (NO)
OS_GRANTED	VARCHAR2 (3)		Granted by the operating system (YES) or not (NO); occurs if configuration parameter OS_ROLES = true.

USER_RSRC_CONSUMER_GROUP_PRIVS

USER_RSRC_CONSUMER_GROUP_PRIVS displays information about the resource consumer groups granted to the current user. Its columns (except for GRANTEE) are the same as those in DBA_RSRC_CONSUMER_GROUP_PRIVS.

See Also: ["DBA_RSRC_CONSUMER_GROUP_PRIVS"](#) on page 3-111

USER_RSRC_MANAGER_SYSTEM_PRIVS

USER_RSRC_MANAGER_SYSTEM_PRIVS displays information about the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. Its columns (except for GRANTEE) are the same as those in DBA_RSRC_MANAGER_SYSTEM_PRIVS.

See Also: ["DBA_RSRC_MANAGER_SYSTEM_PRIVS"](#) on page 3-112

USER_RULE_SET_RULES

USER_RULE_SET_RULES describes the rules in the rule sets owned by the current user. Its columns (except for RULE_SET_OWNER) are the same as those in ALL_RULE_SET_RULES.

See Also: ["ALL_RULE_SET_RULES"](#) on page 2-93

USER_RULE_SETS

USER_RULE_SETS describes the rule sets owned by the current user. Its columns (except for RULE_SET_OWNER) are the same as those in ALL_RULE_SETS.

See Also: ["ALL_RULE_SETS"](#) on page 2-94

USER_RULES

USER_RULES describes the rules owned by the current user. Its columns (except for RULE_OWNER) are the same as those in ALL_RULES.

See Also: ["ALL_RULES"](#) on page 2-94

USER_SCHEDULER_JOB_ARGS

USER_SCHEDULER_JOB_ARGS displays information about the arguments of the Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_JOB_ARGS.

See Also: ["ALL_SCHEDULER_JOB_ARGS"](#) on page 2-95

USER_SCHEDULER_JOB_LOG

USER_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs owned by the current user. Its columns are the same as those in ALL_SCHEDULER_JOB_LOG.

See Also: ["ALL_SCHEDULER_JOB_LOG"](#) on page 2-96

USER_SCHEDULER_JOB_RUN_DETAILS

USER_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs owned by the current user. Its columns are the same as those in ALL_SCHEDULER_JOB_RUN_DETAILS.

See Also: ["ALL_SCHEDULER_JOB_RUN_DETAILS"](#) on page 2-97

USER_SCHEDULER_JOBS

USER_SCHEDULER_JOBS displays information about the Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_JOBS.

See Also: ["ALL_SCHEDULER_JOBS"](#) on page 2-97

USER_SCHEDULER_PROGRAM_ARGS

USER_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_PROGRAM_ARGS.

See Also: ["ALL_SCHEDULER_PROGRAM_ARGS"](#) on page 2-99

USER_SCHEDULER_PROGRAMS

USER_SCHEDULER_PROGRAMS displays information about the Scheduler programs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_PROGRAMS.

See Also: ["ALL_SCHEDULER_PROGRAMS"](#) on page 2-100

USER_SCHEDULER_RUNNING_JOBS

USER_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_RUNNING_JOBS.

See Also: ["ALL_SCHEDULER_RUNNING_JOBS"](#) on page 2-101

USER_SCHEDULER_SCHEDULES

USER_SCHEDULER_SCHEDULES displays information about the Scheduler schedules owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_SCHEDULES.

See Also: ["ALL_SCHEDULER_SCHEDULES"](#) on page 2-101

USER_SEC_RELEVANT_COLS

USER_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_SEC_RELEVANT_COLS.

See Also: ["ALL_SEC_RELEVANT_COLS"](#) on page 2-104

USER_SEGMENTS

USER_SEGMENTS describes the storage allocated for the segments owned by the current user's objects. Its columns (except for OWNER, HEADER_FILE, HEADER_BLOCK, and RELATIVE_FNO) are the same as those in DBA_SEGMENTS.

See Also: ["DBA_SEGMENTS"](#) on page 3-117

USER_SEQUENCES

USER_SEQUENCES describes all sequences owned by the current user. Its columns are the same as those in ["ALL_SEQUENCES"](#) on page 2-105.

USER_SOURCE

USER_SOURCE describes the text source of the stored objects owned by the current user. Its columns (except for OWNER) are the same as those in ["ALL_SOURCE"](#) on page 2-106.

USER_SOURCE_TABLES

USER_SOURCE_TABLES describes the existing source tables owned by the current user. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in ALL_SOURCE_TABLES.

See Also: ["ALL_SOURCE_TABLES"](#) on page 2-106

USER_SQLJ_TYPE_ATTRS

USER_SQLJ_TYPE_ATTRS describes the attributes of the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPE_ATTRS.

See Also: ["ALL_SQLJ_TYPE_ATTRS"](#) on page 2-107

USER_SQLJ_TYPE_METHODS

USER_SQLJ_TYPE_METHODS describes the methods of the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPE_METHODS.

See Also: ["ALL_SQLJ_TYPE_METHODS"](#) on page 2-107

USER_SQLJ_TYPES

USER_SQLJ_TYPES describes the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPES.

See Also: ["ALL_SQLJ_TYPES"](#) on page 2-108

USER_SQLSET

USER_SQLSET displays information about the SQL tuning sets owned by the current user. Its columns (except for OWNER) are the same as those in DBA_SQLSET.

See Also: ["DBA_SQLSET"](#) on page 3-120

USER_SQLSET_BINDS

USER_SQLSET_BINDS displays the bind values associated with the SQL tuning sets owned by the current user. Its columns are the same as those in DBA_SQLSET_BINDS.

See Also: ["DBA_SQLSET_BINDS"](#) on page 3-120

USER_SQLSET_REFERENCES

USER_SQLSET_REFERENCES describes whether or not the SQL tuning sets owned by the current user are active. Its columns are the same as those in DBA_SQLSET_REFERENCES.

See Also: ["DBA_SQLSET_REFERENCES"](#) on page 3-120

USER_SQLSET_STATEMENTS

USER_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form the SQL tuning sets owned by the current user. Its columns are the same as those in DBA_SQLSET_STATEMENTS.

See Also: ["DBA_SQLSET_STATEMENTS"](#) on page 3-121

USER_SQLTUNE_BINDS

USER_SQLTUNE_BINDS displays the bind values associated with the tuned SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_BINDS.

See Also: ["DBA_SQLTUNE_BINDS"](#) on page 3-122

USER_SQLTUNE_PLANS

USER_SQLTUNE_PLANS displays information about the execution plans generated for the SQL statements owned by the current user during a SQL tuning session. Its columns are the same as those in DBA_SQLTUNE_PLANS.

See Also: ["DBA_SQLTUNE_PLANS"](#) on page 3-122

USER_SQLTUNE_RATIONALE_PLAN

USER_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of the SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_RATIONALE_PLAN.

See Also: ["DBA_SQLTUNE_RATIONALE_PLAN"](#) on page 3-124

USER_SQLTUNE_STATISTICS

USER_SQLTUNE_STATISTICS displays statistics associated with the SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_STATISTICS.

See Also: ["DBA_SQLTUNE_STATISTICS"](#) on page 3-124

USER_STORED_SETTINGS

USER_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units, but only shows information about PL/SQL units owned by the current user. USER_STORED_SETTINGS does not display the OWNER column. The rest of its columns are the same as those in ["ALL_STORED_SETTINGS"](#) on page 2-109.

USER_SUBPART_COL_STATISTICS

USER_SUBPART_COL_STATISTICS provides column statistics and histogram information for subpartitions of subpartitioned objects owned by the current user. Its columns are the same as those in ["ALL_STORED_SETTINGS"](#) on page 2-109.

USER_SUBPART_HISTOGRAMS

USER_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on table subpartitions owned by the current user. Its columns are the same as those in ["ALL_SUBPART_HISTOGRAMS"](#) on page 2-117.

USER_SUBPART_KEY_COLUMNS

USER_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) owned by the current user. Its columns are the same as those in ALL_SUBPART_KEY_COLUMNS.

See Also: ["ALL_SUBPART_KEY_COLUMNS"](#) on page 2-118

USER_SUBPARTITION_TEMPLATES

USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. Its columns (except for USER_NAME) are the same as those in ALL_SUBPARTITION_TEMPLATES.

See Also: ["ALL_SUBPARTITION_TEMPLATES"](#) on page 2-118

USER_SUBSCRIBED_COLUMNS

USER_SUBSCRIBED_COLUMNS describes the columns of source tables to which the current user has subscribed. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in ALL_SUBSCRIBED_COLUMNS.

See Also: ["ALL_SUBSCRIBED_COLUMNS"](#) on page 2-119

USER_SUBSCRIBED_TABLES

USER_SUBSCRIBED_TABLES describes the source tables to which the current user has subscribed. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in ALL_SUBSCRIBED_TABLES.

See Also: ["ALL_SUBSCRIBED_TABLES"](#) on page 2-119

USER_SUBSCRIPTIONS

USER_SUBSCRIPTIONS describes the subscriptions owned by the current user. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in ALL_SUBSCRIPTIONS.

See Also: ["ALL_SUBSCRIPTIONS"](#) on page 2-120

USER_SYNONYMS

USER_SYNONYMS describes the private synonyms (synonyms owned by the current user). Its columns () are the same as those in ALL_SYNONYMS.

See Also: ["ALL_SYNONYMS"](#) on page 2-121

USER_SYS_PRIVS

USER_SYS_PRIVS lists system privileges granted to the current user. Its columns are the same as those in ["DBA_SYS_PRIVS"](#) on page 3-128.

USER_TAB_COL_STATISTICS

USER_TAB_COL_STATISTICS contains column statistics and histogram information extracted from ["USER_TAB_COLUMNS"](#) on page 3-182. Its columns are the same as those in ["ALL_TAB_COL_STATISTICS"](#) on page 2-121.

USER_TAB_COLS

USER_TAB_COLS describes the columns of the tables, views, and clusters owned by the current user. This view differs from ["USER_TAB_COLUMNS"](#) on page 3-182 in that hidden columns are not filtered out. Its columns (except for OWNER) are the same as those in ["ALL_TAB_COLS"](#) on page 2-122. To gather statistics for this view, use the ANALYZE SQL statement.

USER_TAB_COLUMNS

USER_TAB_COLUMNS describes the columns of the tables, views, and clusters owned by the current user. Its columns (except for OWNER) are the same as those in ["ALL_TAB_COLUMNS"](#) on page 2-124. To gather statistics for this view, use the ANALYZE SQL statement.

USER_TAB_COMMENTS

USER_TAB_COMMENTS displays comments on the tables and views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TAB_COMMENTS.

See Also: ["ALL_TAB_COMMENTS"](#) on page 2-125

USER_TAB_HISTOGRAMS

USER_TAB_HISTOGRAMS describes histograms on columns of tables owned by the current user. Its columns are the same as those in ["ALL_TAB_HISTOGRAMS"](#) on page 2-126.

USER_TAB_MODIFICATIONS

USER_TAB_MODIFICATIONS describes modifications to all tables owned by the current user that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in ["ALL_TAB_MODIFICATIONS"](#) on page 2-126.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time and may not be populated until a few hours after the actual modifications occurred.

USER_TAB_PARTITIONS

USER_TAB_PARTITIONS describes partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE statements for all partitions owned by the current user. Its columns are the same as those in ["ALL_TAB_PARTITIONS"](#) on page 2-127.

USER_TAB_PRIVS

USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee. Its columns are the same as those in DBA_TAB_PRIVS.

See Also: ["DBA_TAB_PRIVS"](#) on page 3-129

USER_TAB_PRIVS_MADE

USER_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner. Its columns (except for OWNER) are the same as those in ALL_TAB_PRIVS_MADE.

See Also: ["ALL_TAB_PRIVS_MADE"](#) on page 2-129

USER_TAB_PRIVS_RECD

USER_TAB_PRIVS_RECD describes the object grants for which the current user is the grantee. Its columns (except for GRANTEE) are the same as those in ALL_TAB_PRIVS_RECD.

See Also: ["ALL_TAB_PRIVS_RECD"](#) on page 2-129

USER_TAB_STATISTICS

USER_TAB_STATISTICS displays optimizer statistics for the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TAB_STATISTICS.

See Also: ["ALL_TAB_STATISTICS"](#) on page 2-130

USER_TAB_SUBPARTITIONS

USER_TAB_SUBPARTITIONS describes, for each table subpartition, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in ["ALL_TAB_SUBPARTITIONS"](#) on page 2-131.

Note: Statistics are not collected on a per-subpartition basis.

USER_TABLES

USER_TABLES describes the relational tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TABLES. To gather statistics for this view, use the ANALYZE SQL statement.

See Also: ["ALL_TABLES"](#) on page 2-132

USER_TABLESPACES

USER_TABLESPACES describes the tablespaces accessible to the current user. Its columns (except for PLUGGED_IN) are the same as those in DBA_TABLESPACES.

See Also: ["DBA_TABLESPACES"](#) on page 3-130

USER_TRANSFORMATIONS

USER_TRANSFORMATIONS displays information about the transformations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_TRANSFORMATIONS.

See Also: ["DBA_TRANSFORMATIONS"](#) on page 3-133

USER_TRIGGER_COLS

USER_TRIGGER_COLS describes the use of columns in triggers owned by the current user and triggers on tables owned by the current user. Its columns are the same as those in ["ALL_TRIGGER_COLS"](#) on page 2-134.

USER_TRIGGERS

USER_TRIGGERS describes all triggers owned by the current. Its columns are the same as those in ["ALL_TRIGGERS"](#) on page 2-135.

USER_TS_QUOTAS

USER_TS_QUOTAS contains information about tablespace quotas for the current user. Its columns are the same as those in ["DBA_TS_QUOTAS"](#) on page 3-133.

USER_TUNE_MVIEW

USER_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure. Its columns (except for OWNER) are the same as those in DBA_TUNE_MVIEW.

See Also: ["DBA_TUNE_MVIEW"](#) on page 3-134

USER_TYPE_ATTRS

USER_TYPE_ATTRS describes the attributes of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_ATTRS.

See Also: ["ALL_TYPE_ATTRS"](#) on page 2-135

USER_TYPE_METHODS

USER_TYPE_METHODS describes the methods of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_METHODS.

See Also: ["ALL_TYPE_METHODS"](#) on page 2-136

USER_TYPE_VERSIONS

USER_TYPE_VERSIONS describes the versions of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_VERSIONS.

See Also: ["ALL_TYPE_VERSIONS"](#) on page 2-137

USER_TYPES

USER_TYPES describes the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPES.

See Also: ["ALL_TYPES"](#) on page 2-137

USER_UNUSED_COL_TABS

USER_UNUSED_COL_TABS contains a description of all tables containing unused columns. Its columns are the same as those in ["ALL_UNUSED_COL_TABS"](#) on page 2-138.

USER_UPDATABLE_COLUMNS

USER_UPDATABLE_COLUMNS describes columns in a join view that can be updated by the current user. Its columns are the same as those in ["ALL_UPDATABLE_COLUMNS"](#) on page 2-138.

See Also: *Oracle Database Concepts* for information on updatable join views

USER_USERS

USER_USERS describes the current user. Its columns (except for PASSWORD and PROFILE) are the same as those in DBA_USERS.

See Also: ["DBA_USERS"](#) on page 3-135

USER_USTATS

USER_USTATS describes all user-defined statistics owned by the current user. Its columns are the same as those in "ALL_USTATS" on page 2-139.

USER_VARRAYS

USER_VARRAYS describes the varrays owned by the current user. Its columns are the same as those in "ALL_VARRAYS" on page 2-140.

USER_VIEWS

USER_VIEWS describes the views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_VIEWS.

See Also: "ALL_VIEWS" on page 2-140

USER_WARNING_SETTINGS

USER_WARNING_SETTINGS displays information about the warning parameter settings for the objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_WARNING_SETTINGS.

See Also: "ALL_WARNING_SETTINGS" on page 2-141

USER_XML_SCHEMAS

USER_XML_SCHEMAS describes the registered XML schemas owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_SCHEMAS.

See Also: "ALL_XML_SCHEMAS" on page 2-142

USER_XML_TAB_COLS

USER_XML_TAB_COLS describes the columns of the XML tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_TAB_COLS.

See Also: "ALL_XML_TAB_COLS" on page 2-142

USER_XML_TABLES

USER_XML_TABLES describes the XML tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_TABLES.

See Also: "ALL_XML_TABLES" on page 2-143

USER_XML_VIEW_COLS

USER_XML_VIEW_COLS describes the columns of the XML views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_VIEW_COLS.

See Also: "ALL_XML_VIEW_COLS" on page 2-143

USER_XML_VIEWS

USER_XML_VIEWS describes the XML views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_VIEWS.

See Also: ["ALL_XML_VIEWS"](#) on page 2-144

Part III

Dynamic Performance Views

This part describes the dynamic performance views, which are often referred to as V\$ views. This part contains the following chapters:

- [Chapter 4, "Dynamic Performance \(V\\$\) Views"](#)
- [Chapter 5, "Dynamic Performance \(V\\$\) Views: V\\$NLS_PARAMETERS to V\\$WAITSTAT"](#)

Dynamic Performance (V\$) Views

This chapter describes (in alphabetical order) dynamic performance views.

This chapter contains the following topics:

- [About Dynamic Performance Views](#)
- [Dynamic Performance View Descriptions](#)

About Dynamic Performance Views

Oracle contains a set of underlying views that are maintained by the database server and accessible to the database administrator user `SYS`. These views are called **dynamic performance views** because they are continuously updated while a database is open and in use, and their contents relate primarily to performance.

Although these views appear to be regular database tables, they are not. These views provide data on internal disk structures and memory structures. You can select from these views, but you can never update or alter them.

Note:

- You can query the dynamic performance views to extract information from them. However, only simple queries are supported. If sorts, joins, `GROUP BY` clauses and the like are needed, then you should copy the information from each V\$ view into a table (for example, using a `CREATE TABLE . . . AS SELECT` statement), and then query from those tables.
 - Because the information in the V\$ views is dynamic, read consistency is not guaranteed for `SELECT` operations on these views.
-
-

The `catalog.sql` script contains definitions of the views and public synonyms for the dynamic performance views. You must run `catalog.sql` to create these views and synonyms. After installation, only user `SYS` or anyone with `SYSDBA` role has access to the dynamic performance tables.

V\$ Views

The actual dynamic performance views are identified by the prefix `V_`. Public synonyms for these views have the prefix `V$`. Database administrators and other users should access only the `V$` objects, not the `V_` objects.

The dynamic performance views are used by Oracle Enterprise Manager, which is the primary interface for accessing information about system performance. After an instance is started, the V\$ views that read from memory are accessible. Views that read data from disk require that the database be mounted, and some require that the database be open.

GV\$ Views

For almost every V\$ view described in this chapter, Oracle has a corresponding GV\$ (global V\$) view. In Real Application Clusters, querying a GV\$ view retrieves the V\$ view information from all qualified instances. In addition to the V\$ information, each GV\$ view contains an extra column named INST_ID of datatype NUMBER. The INST_ID column displays the instance number from which the associated V\$ view information was obtained. The INST_ID column can be used as a filter to retrieve V\$ information from a subset of available instances. For example, the following query retrieves the information from the V\$LOCK view on instances 2 and 5:

```
SQL> SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5;
```

In order to query the GV\$ views, the value of the PARALLEL_MAX_SERVERS initialization parameter must be greater than zero on all instances mounting the database.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

Dynamic Performance View Descriptions

The remainder of this chapter describes the dynamic performance views in alphabetical order.

V\$ACCESS

V\$ACCESS displays objects in the database that are currently locked and the sessions that are accessing them.

Column	Datatype	Description
SID	NUMBER	Session number that is accessing an object
OWNER	VARCHAR2(64)	Owner of the object
OBJECT	VARCHAR2(1000)	Name of the object
TYPE	VARCHAR2(24)	Type identifier for the object

V\$ACTIVE_INSTANCES

V\$ACTIVE_INSTANCES displays the mapping between instance names and instance numbers for all instances that have the database currently mounted.

Column	Datatype	Description
INST_NUMBER	NUMBER	Instance number
INST_NAME	VARCHAR2(60)	Instance name

V\$ACTIVE_SERVICES

V\$ACTIVE_SERVICES displays the active services in the database.

Column	Datatype	Description
SERVICE_ID	NUMBER	Service ID
NAME	VARCHAR2 (64)	Name of the service
NAME_HASH	NUMBER	Hash of the service name
NETWORK_NAME	VARCHAR2 (512)	Network name
CREATION_DATE	DATE	Creation date
CREATION_DATE_HASH	NUMBER	Creation date hash

V\$ACTIVE_SESS_POOL_MTH

V\$ACTIVE_SESS_POOL_MTH displays available active session pool resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2 (40)	Name of the active session pool resource allocation method

V\$ACTIVE_SESSION_HISTORY

V\$ACTIVE_SESSION_HISTORY displays sampled session activity in the database. It contains snapshots of active database sessions taken once a second. A database session is considered active if it was on the CPU or was waiting for an event that didn't belong to the Idle wait class. Refer to the V\$EVENT_NAME view for more information on wait classes.

This view contains one row for each active session per sample and returns the latest session sample rows first. A majority of the columns describing the session in the active session history are present in the V\$SESSION view.

Column	Datatype	Description
SAMPLE_ID	NUMBER	ID of the sample
SAMPLE_TIME	TIMESTAMP (3)	Time at which the sample was taken
SESSION_ID	NUMBER	Session identifier; maps to V\$SESSION . SID
SESSION_SERIAL#	NUMBER	Session serial number (used to uniquely identify a session's objects); maps to V\$SESSION . SERIAL#
USER_ID	NUMBER	Oracle user identifier; maps to V\$SESSION . USER#
SQL_ID	VARCHAR2 (13)	SQL identifier of the SQL statement that the session was executing at the time of sampling
SQL_CHILD_NUMBER	NUMBER	Child number of the SQL statement that the session was executing at the time of sampling
SQL_PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for the cursor. This information might not be available for all session samples. V\$SESSION does not contain this information.
SQL_OPCODE	NUMBER	Indicates what phase of operation the SQL statement was in; maps to V\$SESSION . COMMAND See Also: "V\$SESSION" on page 5-37 for information on interpreting this column
SERVICE_HASH	NUMBER	Hash that identifies the Service; maps to V\$ACTIVE_SERVICES . NAME_HASH

Column	Datatype	Description
SESSION_TYPE	VARCHAR2(10)	Session type: <ul style="list-style-type: none"> ■ FOREGROUND ■ BACKGROUND
SESSION_STATE	VARCHAR2(7)	Session state: <ul style="list-style-type: none"> ■ WAITING ■ ON CPU
QC_SESSION_ID	NUMBER	Query coordinator session ID. This information is only available if the sampled session is a parallel query slave. For all other sessions, the value is 0.
QC_INSTANCE_ID	NUMBER	Query coordinator instance ID. This information is only available if the sampled session is a parallel query slave. For all other sessions, the value is 0.
EVENT	VARCHAR2(64)	If SESSION_STATE = WAITING, then the event for which the session was waiting for at the time of sampling. If SESSION_STATE = ON CPU, then the event for which the session last waited upon before being sampled. See Also: Appendix C, "Oracle Wait Events"
EVENT_ID	NUMBER	Identifier of the resource or event for which the session is waiting or for which the session last waited. Interpretation is similar to that of the EVENT column.
EVENT#	NUMBER	Number of the resource or event for which the session is waiting or for which the session last waited. Interpretation is similar to that of the EVENT column.
SEQ#	NUMBER	Sequence number that uniquely identifies the wait (incremented for each wait)
P1	NUMBER	First additional parameter
P2	NUMBER	Second additional parameter
P3	NUMBER	Third additional parameter
WAIT_TIME	NUMBER	0 if the session was waiting at the time of sampling Total wait time for the event for which the session last waited if the session was on the CPU when sampled Whether or not WAIT_TIME = 0 is what is useful to find the SESSION_STATE at the time of sampling, rather than the actual value of WAIT_TIME itself. Maps to V\$SESSION.WAIT_TIME.
TIME_WAITED	NUMBER	If SESSION_STATE = WAITING, then the time that the session actually spent waiting for that EVENT. This column is set for waits that were in progress at the time the sample was taken. If a wait event lasted for more than a second and was caught waiting in more than one session sample row, then the actual time spent waiting for that wait event will be populated in the last of those session sample rows. At any given time, this information will not be available for the latest session sample.
CURRENT_OBJ#	NUMBER	Object ID of the object that the session is referencing. This information is only available if the session is waiting for some I/O-related events or for some enqueue waits. Maps to V\$SESSION.ROW_WAIT_OBJ#.
CURRENT_FILE#	NUMBER	File number of the file containing the block that the session is referencing. This information is only available if the session was waiting for some I/O-related events or for some enqueue waits. Maps to V\$SESSION.ROW_WAIT_FILE#.
CURRENT_BLOCK#	NUMBER	ID of the block that the session is referencing. This information is only available if the session was waiting for some I/O-related events or for some enqueue waits. Maps to V\$SESSION.ROW_WAIT_BLOCK#.
PROGRAM	VARCHAR2(48)	Name of the operating system program
MODULE	VARCHAR2(48)	Name of the executing module when sampled, as set by the DBMS_APPLICATION_INFO.SET_MODULE procedure

Column	Datatype	Description
ACTION	VARCHAR2 (32)	Name of the executing module when sampled, as set by the DBMS_APPLICATION_INFO.SET_ACTION procedure
CLIENT_ID	VARCHAR2 (64)	Client identifier of the session; maps to V\$SESSION.CLIENT_IDENTIFIER

V\$ALERT_TYPES

V\$ALERT_TYPES displays information about server alert types.

Column	Datatype	Description
REASON_ID	NUMBER	ID of the alert reason
OBJECT_TYPE	VARCHAR2 (64)	Object type
TYPE	VARCHAR2 (9)	Alert type: <ul style="list-style-type: none"> ▪ Stateful ▪ Stateless
GROUP_NAME	VARCHAR2 (64)	Group name
SCOPE	VARCHAR2 (8)	Scope: <ul style="list-style-type: none"> ▪ Database ▪ Instance
INTERNAL_METRIC_CATEGORY	VARCHAR2 (64)	Internal metric category
INTERNAL_METRIC_NAME	VARCHAR2 (64)	Internal metric name

V\$AQ

V\$AQ displays statistics for the queues in the database.

Column	Datatype	Description
QID	NUMBER	Unique queue identifier
WAITING	NUMBER	Number of messages in the queue in the state 'WAITING'
READY	NUMBER	Number of messages in the queue in the state 'READY'
EXPIRED	NUMBER	Number of messages in the queue the state 'EXPIRED'
TOTAL_WAIT	NUMBER	Total wait time of all 'READY' messages in the queue
AVERAGE_WAIT	NUMBER	Average wait time of 'READY' messages in the queue

V\$ARCHIVE

V\$ARCHIVE displays information about redo log files in need of archiving. Each row displays information for one thread. This information is also available in V\$LOG. Oracle recommends that you use V\$LOG.

See Also: ["V\\$LOG"](#) on page 4-78

Column	Datatype	Description
GROUP#	NUMBER	Log file group number
THREAD#	NUMBER	Log file thread number
SEQUENCE#	NUMBER	Log file sequence number
ISCURRENT	VARCHAR2 (3)	This is the current online redo log

V\$ARCHIVE_DEST

Column	Datatype	Description
CURRENT	VARCHAR2(3)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in ISCURRENT.
FIRST_CHANGE#	NUMBER	First SCN stored in the current log

V\$ARCHIVE_DEST

V\$ARCHIVE_DEST displays, for the current instance, all of the destinations in the Data Guard configuration, including each destination's current value, mode, and status.

Column	Datatype	Description
DEST_ID	NUMBER	Log archive destination parameter identifier (1 to 10)
DEST_NAME	VARCHAR2(256)	Log archive destination parameter name
STATUS	VARCHAR2(9)	Identifies the current status of the destination: <ul style="list-style-type: none">VALID - Initialized and availableINACTIVE - No destination informationDEFERRED - Manually disabled by the userERROR - Error during open or copyDISABLED - Disabled after errorBAD_PARAM - Parameter has errorsALTERNATE - Destination is in an alternate stateFULL - Exceeded quota size for the destination
BINDING	VARCHAR2(9)	Specifies how failure will affect the archival operation: <ul style="list-style-type: none">MANDATORY - Successful archival is requiredOPTIONAL - Successful archival is not required (depends on LOG_ARCHIVE_MIN_SUCCEED_DEST)
NAME_SPACE	VARCHAR2(7)	Identifies the scope of parameter setting: <ul style="list-style-type: none">SYSTEM - System definitionSESSION - Session definition
TARGET	VARCHAR2(7)	Specifies whether the archive destination is local or remote to the primary database: <ul style="list-style-type: none">PRIMARY - localSTANDBY - remote
ARCHIVER	VARCHAR2(10)	Identifies the archiver process relative to the database where the query is issued: <ul style="list-style-type: none">ARCnBACKGROUNDLGWRRFS
SCHEDULE	VARCHAR2(8)	Indicates whether the archival of this destination is INACTIVE, PENDING, ACTIVE, or LATENT
DESTINATION	VARCHAR2(256)	Specifies the location where the archived redo logs are to be archived
LOG_SEQUENCE	NUMBER	Identifies the sequence number of the last archived redo log to be archived
REOPEN_SECS	NUMBER	Identifies the retry time (in seconds) after error
DELAY_MINS	NUMBER	Identifies the delay interval (in minutes) before the archived redo log is automatically applied to a standby database
NET_TIMEOUT	NUMBER	Number of seconds the log writer process will wait for status from the network server of a network operation issued by the log writer process

Column	Datatype	Description
PROCESS	VARCHAR2(10)	Identifies the archiver process relative to the primary database, even if the query is issued on the standby database: <ul style="list-style-type: none"> ■ ARCn ■ FOREGROUND ■ LGWR
REGISTER	VARCHAR2(3)	Indicates whether the archived redo log is registered in the remote destination control file (YES) or not (NO). If the archived redo log is registered, it is available to log apply services.
FAIL_DATE	DATE	Date and time of last error
FAIL_SEQUENCE	NUMBER	Sequence number of the archived redo log being archived when the last error occurred
FAIL_BLOCK	NUMBER	Block number of the archived redo log being archived when the last error occurred
FAILURE_COUNT	NUMBER	Current number of contiguous archival operation failures that have occurred for the destination
MAX_FAILURE	NUMBER	Allows you to control the number of times log transport services will attempt to reestablish communication and resume archival operations with a failed destination
ERROR	VARCHAR2(256)	Displays the error text
ALTERNATE	VARCHAR2(256)	Alternate destination, if any
DEPENDENCY	VARCHAR2(256)	Dependent archive destination, if any
REMOTE_TEMPLATE	VARCHAR2(256)	Specifies the template to be used to derive the location to be recorded
QUOTA_SIZE	NUMBER	Destination quotas, expressed in bytes
QUOTA_USED	NUMBER	Size of all the archived redo logs currently residing on the specified destination
MOUNTID	NUMBER	Instance mount identifier
TRANSMIT_MODE	VARCHAR2(12)	Specifies network transmission mode: <ul style="list-style-type: none"> ■ SYNC=PARALLEL ■ SYNC=NOPARALLEL ■ ASYNC
ASYNC_BLOCKS	NUMBER	Number of blocks specified for the ASYNC attribute
AFFIRM	VARCHAR2(3)	Specifies disk I/O mode
TYPE	VARCHAR2(7)	Indicates whether the archived log destination definition is PUBLIC or PRIVATE. Only PUBLIC destinations can be modified at runtime using the ALTER SYSTEM SET or ALTER SESSION SET statements. By default, all archived log destinations are PUBLIC.
VALID_NOW	VARCHAR2(16)	Indicates whether the destination is valid right now for archival operations: <ul style="list-style-type: none"> ■ YES - Redo log type and database role for this destination are valid for the current database ■ WRONG VALID_TYPE - Redo log type specified for this destination is not valid for the current database role. For example, WRONG VALID_TYPE would be returned if a destination specified with the VALID_FOR=(STANDBY_LOGFILE,STANDBY_ROLE) attribute is running in the standby database role but does not have standby redo logs implemented. ■ WRONG VALID_ROLE - Database role specified for this destination is not the role in which the database is currently running. For example, the WRONG VALID_ROLE would be returned when a destination defined with the VALID_FOR=(ONLINE_LOGFILE,STANDBY_ROLE) attribute is running in the primary database role. ■ INACTIVE - Destination is inactive, probably due to an error

V\$ARCHIVE_DEST_STATUS

Column	Datatype	Description
VALID_TYPE	VARCHAR2(15)	Redo log type or types that are valid for the destination: <ul style="list-style-type: none">▪ ONLINE_LOGFILE▪ STANDBY_LOGFILE▪ ALL_LOGFILES
VALID_ROLE	VARCHAR2(12)	Database role or roles that are valid for the destination: <ul style="list-style-type: none">▪ PRIMARY_ROLE▪ STANDBY_ROLE▪ ALL_ROLES
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name
VERIFY	VARCHAR2(3)	Indicates whether the value of the VERIFY attribute on the LOG_ARCHIVE_DEST_n parameter is verified (YES) or not verified (NO)

See Also:

- ["LOG_ARCHIVE_DEST"](#) on page 1-70 and ["LOG_ARCHIVE_DEST_n"](#) on page 1-71
- ["LOG_ARCHIVE_DUPLEX_DEST"](#) on page 1-73 and ["LOG_ARCHIVE_DEST_STATE_n"](#) on page 1-72
- ["STANDBY_ARCHIVE_DEST"](#) on page 1-134
- ["LOG_ARCHIVE_MIN_SUCCEED_DEST"](#) on page 1-76

V\$ARCHIVE_DEST_STATUS

V\$ARCHIVE_DEST_STATUS displays runtime and configuration information for the archived redo log destinations. The information in this view does not persist across an instance shutdown.

Column	Datatype	Description
DEST_ID	NUMBER	Identifies the log archive destination parameter (1 to 10)
DEST_NAME	VARCHAR2(256)	Log archive destination parameter name
STATUS	VARCHAR2(9)	Current status of the destination: <ul style="list-style-type: none">▪ VALID - Initialized and available▪ INACTIVE - No destination information▪ DEFERRED - Manually disabled by the user▪ ERROR - Error during open or copy▪ DISABLED - Disabled after error▪ BAD_PARAM - Parameter has errors▪ ALTERNATE - Destination is in an alternate state▪ FULL - Exceeded quota size for the destination
TYPE	VARCHAR2(14)	Type of archival destination database: <ul style="list-style-type: none">▪ LOCAL - Local to primary database▪ PHYSICAL - Physical standby▪ CROSS-INSTANCE - An instance of the primary

Column	Datatype	Description
DATABASE_MODE	VARCHAR2(15)	Current mode of the archival destination database: <ul style="list-style-type: none"> STARTED - Instance started, not mounted MOUNTED - Mounted MOUNTED-STANDBY - Mounted standby OPEN - Open read/write OPEN_READ-ONLY - Open read-only
RECOVERY_MODE	VARCHAR2(23)	Current mode of media recovery at the archival destination database: <ul style="list-style-type: none"> IDLE - Managed recovery is not active MANUAL - Manual media recovery active MANAGED - Managed recovery is active MANAGED REAL TIME APPLY - Log apply services recover redo data from standby redo logs at the same time the logs are being written to, as opposed to recovering redo from archived redo logs when a log switch occurs
PROTECTION_MODE	VARCHAR2(20)	Indicates whether the database is protected: <ul style="list-style-type: none"> MAXIMUM PROTECTION MAXIMUM AVAILABILITY RESYNCHRONIZATION MAXIMUM PERFORMANCE UNPROTECTED
DESTINATION	VARCHAR2(256)	Specifies the location where the redo data is to be archived
STANDBY_LOGFILE_COUNT	NUMBER	Indicates the total number of standby redo logs created on the standby database
STANDBY_LOGFILE_ACTIVE	NUMBER	Indicates the total number of standby redo logs on the standby database that are active and contain primary database online redo log information
ARCHIVED_THREAD#	NUMBER	Identifies the thread number of the most recent archived redo log received at the destination
ARCHIVED_SEQ#	NUMBER	Identifies the log sequence number of the most recent archived redo log received at the destination
APPLIED_THREAD#	NUMBER	Identifies the thread number of the most recent applied redo log received at the destination
APPLIED_SEQ#	NUMBER	Identifies the log sequence number of the most recent applied redo log received at the destination
ERROR	VARCHAR2(256)	Displays the error text
SRL	VARCHAR2(3)	Indicates whether standby redo logfiles are used on the standby database (YES) or not (NO)
DB_UNIQUE_NAME	VARCHAR2(30)	Specifies the unique database name of the current instance that was defined with the DB_UNIQUE_NAME attribute on the LOG_ARCHIVE_DEST_n parameter

V\$ARCHIVE_GAP

V\$ARCHIVE_GAP displays information about archive gaps on a standby database. This view can be used to find out the current archive gap that is blocking recovery.

Column	Datatype	Description
THREAD#	NUMBER	Thread number
LOW_SEQUENCE#	NUMBER	Low number of the log file
HIGH_SEQUENCE#	NUMBER	High number of the log file

V\$ARCHIVE_PROCESSES

V\$ARCHIVE_PROCESSES displays the state of the various ARCH processes for the instance.

Column	Datatype	Description
PROCESS	NUMBER	Identifier for the ARCH process for the instance, numbered from 0-9
STATUS	VARCHAR2(10)	Status of the ARCH process, displayed as a keyword. Possible values are: STOPPED, SCHEDULED, STARTING, ACTIVE, STOPPING, and TERMINATED.
LOG_SEQUENCE	NUMBER	This is the online redo log sequence number currently being archived, if STATE="BUSY"
STATE	VARCHAR2(4)	This is the current state of the ARCH process, displayed as a keyword. Possible keywords are IDLE or BUSY.

V\$ARCHIVED_LOG

V\$ARCHIVED_LOG displays archived log information from the control file, including archive log names. An archive log record is inserted after the online redo log is successfully archived or cleared (name column is NULL if the log was cleared). If the log is archived twice, there will be two archived log records with the same THREAD#, SEQUENCE#, and FIRST_CHANGE#, but with a different name. An archive log record is also inserted when an archive log is restored from a backup set or a copy and whenever a copy of a log is made with the RMAN COPY command.

Column	Datatype	Description
RECID	NUMBER	Archived log record ID
STAMP	NUMBER	Archived log record stamp
NAME	VARCHAR2(513)	Archived log file name. If set to NULL, the log file was cleared before it was archived.
DEST_ID	NUMBER	Original destination from which the archivelog was generated. The value is 0 if the destination identifier is not available.
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# of the database when the log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when the log was written
RESETLOGS_ID	NUMBER	Resetlogs identifier associated with the archived redo log
FIRST_CHANGE#	NUMBER	First change# in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change in the next log
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log (in blocks)
BLOCK_SIZE	NUMBER	Redo log block size. This is the logical block size of the archived log, which is the same as the logical block size of the online log from which the archived log was copied. The online log logical block size is a platform-specific value that is not adjustable by the user.

Column	Datatype	Description
CREATOR	VARCHAR2 (7)	Creator of the archivelog: <ul style="list-style-type: none"> ■ ARCH - ARCH process ■ FGRD - Foreground process ■ RMAN ■ SRMN - RMAN at standby ■ LGWR - LGWR process
REGISTRAR	VARCHAR2 (7)	Registrar of the entry: <ul style="list-style-type: none"> ■ RFS ■ ARCH - ARCH process ■ FGRD - Foreground process ■ RMAN ■ SRMN - RMAN at standby ■ LGWR - LGWR process
STANDBY_DEST	VARCHAR2 (3)	Indicates whether the entry is an archivelog destination (YES) or not (NO)
ARCHIVED	VARCHAR2 (3)	Indicates whether the online redo log was archived (YES) or whether RMAN only inspected the log and created a record for future application of redo logs during recovery (NO). See Also: <i>Oracle Database Backup and Recovery Basics</i> .
APPLIED	VARCHAR2 (3)	Indicates whether the archivelog has been applied to its corresponding standby database (YES) or not (NO). The value is always NO for local destinations. This column is meaningful at the physical standby site for the ARCHIVED_LOG entries with REGISTRAR='RFS' (which means this log is shipped from the primary to the standby database). If REGISTRAR='RFS' and APPLIED is NO, then the log has arrived at the standby but has not yet been applied. If REGISTRAR='RFS' and APPLIED is YES, the log has arrived and been applied at the standby database. You can use this field to identify archivelogs that can be backed up and removed from disk.
DELETED	VARCHAR2 (3)	Indicates whether an RMAN DELETE command has physically deleted the archived log file from disk, as well as logically removing it from the control file of the target database and from the recovery catalog (YES) or not (NO)
STATUS	VARCHAR2 (1)	Status of the archived log: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the archiving completed
DICTIONARY_BEGIN	VARCHAR2 (3)	Indicates whether the log contains the start of a LogMiner dictionary (YES) or not (NO)
DICTIONARY_END	VARCHAR2 (3)	Indicates whether the log contains the end of a LogMiner dictionary (YES) or not (NO)
END_OF_REDO	VARCHAR2 (3)	Indicates whether the archived redo log contains the end of all redo information from the primary database (YES) or not (NO)
BACKUP_COUNT	NUMBER	Indicates the number of times this file has been backed up. Values range from 0-15. If the file has been backed up more than 15 times, the value remains 15.
ARCHIVAL_THREAD#	NUMBER	Redo thread number of the instance that performed the archival operation. This column differs from the THREAD# column only when a closed thread is archived by another instance.
ACTIVATION#	NUMBER	Number assigned to the database instantiation
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)

Column	Datatype	Description
FAL	VARCHAR2 (3)	Indicates whether the archive log was generated as the result of a FAL request (YES) or not (NO)

V\$ASM_ALIAS

In an Automatic Storage Management instance, V\$ASM_ALIAS displays one row for every alias present in every disk group mounted by the Automatic Storage Management instance. In a database instance, V\$ASM_ALIAS displays no rows.

Column	Datatype	Description
NAME	VARCHAR2 (48)	Automatic Storage Management alias or alias directory name
GROUP_NUMBER	NUMBER	Owning disk group number of the alias (foreign key to the V\$ASM_DISKGROUP view)
FILE_NUMBER	NUMBER	Automatic Storage Management file number of the alias (foreign key to the V\$ASM_FILE view)
FILE_INCARNATION	NUMBER	Automatic Storage Management file incarnation number for the alias
ALIAS_INDEX	NUMBER	Alias entry number for the alias
ALIAS_INCARNATION	NUMBER	Incarnation number for the parent of the alias
PARENT_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and an alias entry number in the low-order 24 bits (number of the directory containing the alias)
REFERENCE_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and an alias entry number in the low-order 24 bits (number of the directory describing the current entry). For alias entries, the REFERENCE_INDEX is set to 0.
ALIAS_DIRECTORY	VARCHAR2 (1)	Indicates whether the alias is to a directory (Y) or to an Automatic Storage Management file (N)
SYSTEM_CREATED	VARCHAR2 (1)	Indicates whether the alias is system created (Y) or user created (N)

V\$ASM_CLIENT

In an Automatic Storage Management instance, V\$ASM_CLIENT displays one row for every database instance using a disk group managed by the Automatic Storage Management instance. In a database instance, V\$ASM_CLIENT displays one row for the Automatic Storage Management instance if the database has open Automatic Storage Management files.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group being used by the client database instance (foreign key to the V\$ASM_DISKGROUP view)
INSTANCE_NAME	VARCHAR2 (64)	Identifier for the database instance client
DB_NAME	VARCHAR2 (8)	Unique database name of the database client instance
STATUS	VARCHAR2 (12)	Status of the client connection: <ul style="list-style-type: none"> ■ CONNECTED - Database instance client has an active connection to the Automatic Storage Management instance ■ DISCONNECTED - Database instance client normally ended its connection to the Automatic Storage Management instance ■ BROKEN - Connection with the database instance client terminated abnormally

V\$ASM_DISK

In an Automatic Storage Management instance, V\$ASM_DISK displays one row for every disk discovered by the Automatic Storage Management instance, including disks which are not part of any disk group. In a database instance, V\$ASM_DISK only displays rows for disks in disk groups in use by the database instance.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group containing the disk (foreign key to the V\$ASM_DISKGROUP view)
DISK_NUMBER	NUMBER	Number assigned to the disk within its disk group
COMPOUND_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and a disk number in the low-order 24 bits (for efficient access to the view)
INCARNATION	NUMBER	Incarnation number for the disk
MOUNT_STATUS	VARCHAR2 (7)	Per-instance status of the disk relative to group mounts: <ul style="list-style-type: none"> ▪ MISSING - Automatic Storage Management metadata indicates that the disk is known to be part of the Automatic Storage Management disk group, but no disk in the storage system was found with the indicated name ▪ CLOSED - Disk is present in the storage system but is not being accessed by Automatic Storage Management ▪ OPENED - Disk is present in the storage system and is being accessed by Automatic Storage Management. This is the normal state for disks in a database instance which are part of a Disk Group being actively used by the instance. ▪ CACHED - Disk is present in the storage system, and is part of a disk group being accessed by the Automatic Storage Management instance. This is the normal state for disks in an Automatic Storage Management instance which are part of a mounted disk group.
HEADER_STATUS	VARCHAR2 (12)	Per-instance status of the disk as seen by discovery: <ul style="list-style-type: none"> ▪ UNKNOWN - Automatic Storage Management disk header has not been read ▪ CANDIDATE - Disk is not part of a disk group and may be added to a disk group with the ALTER DISKGROUP statement ▪ INCOMPATIBLE - Version number in the disk header is not compatible with the Automatic Storage Management software version. ▪ PROVISIONED - Disk is not part of a disk group and may be added to a disk group with the ALTER DISKGROUP statement. The PROVISIONED header status is different from the CANDIDATE header status in that PROVISIONED implies that an additional platform-specific action has been taken by an administrator to make the disk available for Automatic Storage Management. ▪ MEMBER - Disk is a member of an existing disk group. No attempt should be made to add the disk to a different disk group. The ALTER DISKGROUP statement will reject such an addition unless overridden with the FORCE option ▪ FORMER - Disk was once part of a disk group but has been dropped cleanly from the group. It may be added to a new disk group with the ALTER DISKGROUP statement. ▪ CONFLICT - Automatic Storage Management disk was not mounted due to a conflict

Column	Datatype	Description
MODE_STATUS	VARCHAR2 (7)	Global status about which kinds of I/O requests are allowed to the disk: <ul style="list-style-type: none"> ■ UNKNOWN - Automatic Storage Management disk mode is not known (typically the disk is not mounted) ■ ONLINE - Disk is online and operating normally. Reads and writes are attempted. ■ OFFLINE - Disk is offline and access to data is not permitted. Reads and writes are not attempted. An offline disk remains logically part of its disk group.
STATE	VARCHAR2 (8)	Global state of the disk with respect to the disk group: <ul style="list-style-type: none"> ■ UNKNOWN - Automatic Storage Management disk state is not known (typically the disk is not mounted) ■ NORMAL - Disk is online and operating normally ■ ADDING - Disk is being added to a disk group, and is pending validation by all instances that have the disk group mounted ■ DROPPING - Disk has been manually taken offline and space allocation or data access for the disk halts. Rebalancing will commence to relocate data off the disks to other disks in the disk group. Upon completion of the rebalance, the disk is expelled from the group. ■ HUNG - Disk drop operation cannot continue because there is insufficient space to relocate the data from the disk being dropped ■ FORCING - Disk is being removed from the disk group without attempting to offload its data. The data will be recovered from redundant copies, where possible. ■ DROPPED - Disk has been fully expelled from the disk group
REDUNDANCY	VARCHAR2 (7)	External redundancy of the disk: <ul style="list-style-type: none"> ■ UNKNOWN ■ UNPROT ■ MIRROR ■ PARITY
LIBRARY	VARCHAR2 (64)	Name of the library that discovered the disk
TOTAL_MB	NUMBER	Total capacity of the disk (in megabytes)
FREE_MB	NUMBER	Unused capacity of the disk (in megabytes)
NAME	VARCHAR2 (30)	Name of the disk
FAILGROUP	VARCHAR2 (30)	Name of the failure group containing the disk
LABEL	VARCHAR2 (31)	Disk label portion of the name returned by discovery
PATH	VARCHAR2 (256)	Operating system pathname portion of the name returned by discovery
UDID	VARCHAR2 (64)	Universal Device ID portion of the name returned by discovery
CREATE_DATE	DATE	Date and time when the disk was added to the disk group
MOUNT_DATE	DATE	Date and time when the disk was mounted by the first instance
REPAIR_TIMER	NUMBER	Seconds remaining until the disk is automatically dropped (0 if not failed)
READS	NUMBER	Total number of I/O read requests for the disk
WRITES	NUMBER	Total number of I/O write requests for the disk
READ_ERRS	NUMBER	Total number of failed I/O read requests for the disk
WRITE_ERRS	NUMBER	Total number of failed I/O write requests for the disk
READ_TIME	NUMBER	Total I/O time (in hundredths of a second) for read requests for the disk if the TIMED_STATISTICS initialization parameter is set to true (0 if set to false)
WRITE_TIME	NUMBER	Total I/O time (in hundredths of a second) for write requests for the disk if the TIMED_STATISTICS initialization parameter is set to true (0 if set to false)
BYTES_READ	NUMBER	Total number of bytes read from the disk

Column	Datatype	Description
BYTES_WRITTEN	NUMBER	Total number of bytes written to the disk

Note: The GROUP_NUMBER and DISK_NUMBER columns will only be valid if the disk is part of a disk group which is currently mounted by the instance. Otherwise, GROUP_NUMBER will be 0, and DISK_NUMBER will be a unique value with respect to the other disks that also have a group number of 0.

V\$ASM_DISKGROUP

In an Automatic Storage Management instance, V\$ASM_DISKGROUP displays one row for every disk group discovered by the Automatic Storage Management instance. In a database instance, V\$ASM_DISKGROUP displays one row for every Automatic Storage Management disk group mounted by the local Automatic Storage Management instance.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Cluster-wide number assigned to the disk group (primary key)
NAME	VARCHAR2(30)	Name of the disk group
SECTOR_SIZE	NUMBER	Physical block size (in bytes)
BLOCK_SIZE	NUMBER	Automatic Storage Management metadata block size (in bytes)
ALLOCATION_UNIT_SIZE	NUMBER	Size of the allocation unit (in bytes)
STATE	VARCHAR2(11)	State of the disk group relative to the instance: <ul style="list-style-type: none"> ▪ CONNECTED - Disk group is in use by the database instance ▪ BROKEN - Database instance lost connectivity to the Automatic Storage Management instance that mounted the disk group ▪ UNKNOWN - Automatic Storage Management instance has never attempted to mount the disk group ▪ DISMOUNTED - Disk group was cleanly dismounted by the Automatic Storage Management instance following a successful mount ▪ MOUNTED - Instance is successfully serving the disk group to its database clients
TYPE	VARCHAR2(6)	Redundancy type for the disk group: <ul style="list-style-type: none"> ▪ EXTERN ▪ NORMAL ▪ HIGH
TOTAL_MB	NUMBER	Total capacity of the disk group (in megabytes)
FREE_MB	NUMBER	Unused capacity of the disk group (in megabytes)

Note: The GROUP_NUMBER, TOTAL_MB, and FREE_MB columns are only meaningful if the disk group is mounted by the instance. Otherwise, their values will be 0.

V\$ASM_FILE

In an Automatic Storage Management instance, V\$ASM_FILE displays one row for every Automatic Storage Management file in every disk group mounted by the

Automatic Storage Management instance. In a database instance, V\$ASM_FILE displays no rows.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group containing the file (composite primary key)
FILE_NUMBER	NUMBER	Number of the file within the disk group (composite primary key)
COMPOUND_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and a file number in the low-order 24 bits (for efficient access to the view)
INCARNATION	NUMBER	Incarnation number for the file (composite primary key)
BLOCK_SIZE	NUMBER	Block size of the file (in bytes)
BLOCKS	NUMBER	Number of blocks in the file
BYTES	NUMBER	Number of bytes in the file
SPACE	NUMBER	Number of bytes allocated to the file
TYPE	VARCHAR2(64)	Type of the file
REDUNDANCY	VARCHAR2(6)	Redundancy of the file: <ul style="list-style-type: none"> ▪ UNPROT ▪ MIRROR ▪ PARITY
STRIPED	VARCHAR2(6)	Indicates how the file is striped: <ul style="list-style-type: none"> ▪ FINE ▪ COARSE
CREATION_DATE	DATE	Date on which the file was created
MODIFICATION_DATE	DATE	Date of the last open/close for writing

V\$ASM_OPERATION

In an Automatic Storage Management instance, V\$ASM_OPERATION displays one row for every active Automatic Storage Management long running operation executing in the Automatic Storage Management instance. In a database instance, V\$ASM_OPERATION displays no rows.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Disk group number (primary key). This is the foreign key to the V\$ASM_DISKGROUP view.
OPERATION	VARCHAR2(5)	Type of the operation: <ul style="list-style-type: none"> ▪ REBAL
STATE	VARCHAR2(4)	State of the operation: <ul style="list-style-type: none"> ▪ QUEUED ▪ RUNNING
POWER	NUMBER	Power requested for the operation as specified by the ASM_POWER_LIMIT initialization parameter or command syntax
ACTUAL	NUMBER	Power allocated to the operation
SOFAR	NUMBER	Number of Allocation Units that have been moved so far by the operation
EST_WORK	NUMBER	Estimated number of Allocation Units that have to be moved by the operation
EST_RATE	NUMBER	Estimated number of Allocation Units that are being moved per minute by the operation
EST_MINUTES	NUMBER	Estimated amount of time (in minutes) that the remainder of the operation is expected to take

V\$ASM_TEMPLATE

In an Automatic Storage Management instance, V\$ASM_TEMPLATE displays one row for every template present in every disk group mounted by the Automatic Storage Management instance. In a database instance, V\$ASM_TEMPLATE displays one row for every template present in every disk group mounted by the Automatic Storage Management instance with which the database instance communicates.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Owning disk group number (foreign key to the V\$ASM_DISKGROUP view)
ENTRY_NUMBER	NUMBER	Template number (primary key)
REDUNDANCY	VARCHAR2(6)	Redundancy of the template: <ul style="list-style-type: none"> ▪ UNPROT ▪ MIRROR ▪ PARITY
STRIPE	VARCHAR2(6)	Indicates how the template is striped: <ul style="list-style-type: none"> ▪ FINE ▪ COARSE
SYSTEM	VARCHAR2(1)	Indicates whether the template is a system template (Y) or not (N)
NAME	VARCHAR2(30)	Name of the template

V\$BACKUP

V\$BACKUP displays the backup status of all online datafiles.

Column	Datatype	Description
FILE#	NUMBER	File identifier
STATUS	VARCHAR2(18)	File status: NOT ACTIVE, ACTIVE (backup in progress), OFFLINE NORMAL, or description of an error
CHANGE#	NUMBER	System change number when backup started
TIME	DATE	Time the backup started

V\$BACKUP_ASYNC_IO

V\$BACKUP_ASYNC_IO displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

See Also: *Oracle Database Performance Tuning Guide* for information on how to use this table to tune backup performance

Column	Datatype	Description
SID	NUMBER	Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	Use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp
DEVICE_TYPE	VARCHAR2(17)	Device type where the file is located

Column	Datatype	Description
TYPE	VARCHAR2(9)	INPUT, OUTPUT, or AGGREGATE
STATUS	VARCHAR2(11)	NOT_STARTED, IN_PROGRESS, or FINISHED
FILENAME	VARCHAR2(513)	Name of the backup file being read or written
SET_COUNT	NUMBER	Set count of the backup set being read or written
SET_STAMP	NUMBER	Set stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	Size of the buffers being used to read/write the file, in bytes
BUFFER_COUNT	NUMBER	Number of buffers being used to read/write the file
TOTAL_BYTES	NUMBER	Total number of bytes that will be read or written for the file, if known. If not known, this column will be null
OPEN_TIME	DATE	Time the file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened
CLOSE_TIME	DATE	Time the file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed
ELAPSED_TIME	NUMBER	Time, in hundredths of a second, that the file was open
MAXOPENFILES	NUMBER	Number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'.
BYTES	NUMBER	Number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	Number of I/Os that were performed to this file
READY	NUMBER	Number of asynchronous requests for which a buffer was immediately ready for use
SHORT_WAITS	NUMBER	Number of times that a buffer was not immediately available, but a buffer became available after doing a nonblocking poll for I/O completion
SHORT_WAIT_TIME_TOTAL	NUMBER	Total time, in hundredths of a second, taken by nonblocking polls for I/O completion
SHORT_WAIT_TIME_MAX	NUMBER	Maximum time taken for a nonblocking poll for I/O completion, in hundredths of a second
LONG_WAITS	NUMBER	The number of times that a buffer was not immediately available, and only became available after a blocking wait was issued
LONG_WAIT_TIME_TOTAL	NUMBER	The total time, in hundredths of a second, taken by blocking waits for I/O completion
LONG_WAIT_TIME_MAX	NUMBER	The maximum time taken for a blocking wait for I/O completion, in hundredths of a second

V\$BACKUP_CORRUPTION

V\$BACKUP_CORRUPTION displays information about corrupt block ranges in datafile backups from the control file. Note that corruptions are not tolerated in the control file and archived redo log backups.

Column	Datatype	Description
RECID	NUMBER	Backup corruption record ID
STAMP	NUMBER	Backup corruption record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	backup piece that contains this corrupt block
FILE#	NUMBER	Absolute file number of the datafile that contains the corrupt blocks
BLOCK#	NUMBER	Block number of the first corrupt block in the range of corrupted blocks

Column	Datatype	Description
BLOCKS	NUMBER	Number of corrupted blocks found starting with BLOCK#
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
MARKED_CORRUPT	VARCHAR2 (3)	Indicates whether this corruption was not previously detected by the Oracle Database (YES) or the Oracle Database had already discovered this corrupt block and marked it as corrupt (NO). Note that when a corrupt block is encountered in a backup, and was not already marked corrupt by the Oracle Database, then the backup process does not mark the block as corrupt in the production datafile. Thus, this field may be YES for the same block in more than one backup set.
CORRUPTION_TYPE	VARCHAR2 (9)	Type of block corruption in the datafile: <ul style="list-style-type: none"> ▪ ALL_ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ▪ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ▪ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ▪ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ▪ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V\$BACKUP_DATAFILE

V\$BACKUP_DATAFILE displays information about control files and datafiles in backup sets from the control file.

Column	Datatype	Description
RECID	NUMBER	Backup datafile record ID
STAMP	NUMBER	Backup datafile record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
FILE#	NUMBER	Datafile number; set to 0 for control file
CREATION_CHANGE#	NUMBER	Creation system change number (SCN) of the datafile
CREATION_TIME	DATE	Creation timestamp of the datafile
RESETLOGS_CHANGE#	NUMBER	Resetlogs system change number (SCN) of the datafile when it was backed up
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when it was backed up
INCREMENTAL_LEVEL	NUMBER	(0-4) incremental backup level
INCREMENTAL_CHANGE#	NUMBER	All blocks changed after incremental change# is included in this backup; set to 0 for a full backup
CHECKPOINT_CHANGE#	NUMBER	All changes up to checkpoint change# are included in this backup
CHECKPOINT_TIME	DATE	Timestamp of the checkpoint
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change# in this backup
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt
MEDIA_CORRUPT	NUMBER	Number of blocks media corrupt
LOGICALLY_CORRUPT	NUMBER	Number of blocks logically corrupt

V\$BACKUP_DEVICE

Column	Datatype	Description
DATAFILE_BLOCKS	NUMBER	Size of the datafile in blocks at backup time. This value is also the number of blocks taken by the datafile restarted from this backup.
BLOCKS	NUMBER	Size of the backup datafile (in blocks). Unused blocks are not copied to the backup.
BLOCK_SIZE	NUMBER	Block size
OLDEST_OFFLINE_RANGE	NUMBER	RECID of the oldest offline range record in this backup control file. 0 for datafile backups.
COMPLETION_TIME	DATE	Time completed
CONTROLFILE_TYPE	VARCHAR2(1)	B - Normal copies S - Standby copies
USED_CHANGE_TRACKING	VARCHAR2(3)	Indicates whether change tracking data was used to accelerate this incremental backup (YES) or whether change tracking data was not used (NO)
BLOCKS_READ	NUMBER	Number of blocks that were scanned while taking this backup. If this was an incremental backup, and change tracking was used to optimize the backup, then the value of this column will be smaller than DATAFILE_BLOCKS. Otherwise, the value of this column will be the same as DATAFILE_BLOCKS. Even when change tracking data is used, the value of this column may be larger than BLOCKS, because the data read by change tracking is further refined during the process of creating an incremental backup.

V\$BACKUP_DEVICE

V\$BACKUP_DEVICE displays information about supported backup devices. If a device type does not support named devices, then one row with the device type and a null device name is returned for that device type. If a device type supports named devices then one row is returned for each available device of that type. The special device type DISK is not returned by this view because it is always available.

Column	Datatype	Description
DEVICE_TYPE	VARCHAR2(17)	Type of the backup device
DEVICE_NAME	VARCHAR2(513)	Name of the backup device

V\$BACKUP_FILES

V\$BACKUP_FILES displays information about all RMAN backups (both image copies and backup sets) and archived logs.

This view simulates the LIST BACKUP and LIST COPY RMAN commands. This view requires that the database be set using the DBMS_RCVMAN.SETDATABASE procedure.

Column	Datatype	Description
PKEY	NUMBER	Primary key for the backup
BACKUP_TYPE	VARCHAR2(32)	Type of the backup: <ul style="list-style-type: none">BACKUP SETCOPYPROXY COPY

Column	Datatype	Description
FILE_TYPE	VARCHAR2(32)	Type of the file: <ul style="list-style-type: none"> ■ DATAFILE ■ CONTROLFILE ■ SPFILE ■ REDO LOG ■ COPY (for an image copy backup) ■ PIECE (for a backup piece)
KEEP	VARCHAR2(3)	Indicates whether the backup has a retention policy different from the value for CONFIGURE RETENTION POLICY (YES) or not (NO)
KEEP_UNTIL	DATE	If the KEEP UNTIL TIME clause of the BACKUP command was specified, then this column shows the date after which the backup becomes obsolete. If the column is null and KEEP_OPTIONS is not null, the backup never becomes obsolete.
KEEP_OPTIONS	VARCHAR2(13)	KEEP options for the backup: <ul style="list-style-type: none"> ■ LOGS - RMAN keeps the logs needed to recover the backup ■ NOLOGS - RMAN does not keep the logs needed to recover the backup <p>If this column is null, then the backup has no KEEP options and will be made obsolete based on the retention policy.</p>
STATUS	VARCHAR2(16)	Status of the backup: <ul style="list-style-type: none"> ■ AVAILABLE ■ UNAVAILABLE ■ EXPIRED ■ OTHER
FNAME	VARCHAR2(1024)	Name of the file
TAG	VARCHAR2(32)	Tag of the piece, copy, or proxy copy
MEDIA	VARCHAR2(80)	Media ID of the piece or proxy copy
RECID	NUMBER	Recid of the record in the controlfile
STAMP	NUMBER	Stamp of the record in the controlfile
DEVICE_TYPE	VARCHAR2(255)	Type of media device that stores the backup
BLOCK_SIZE	NUMBER	Block size for the backup (in bytes)
COMPLETION_TIME	DATE	Time when the backup completed
COMPRESSED	VARCHAR2(3)	Indicates whether the backup piece is compressed (YES) or not (NO); valid only if FILE_TYPE is PIECE. Image copies cannot be compressed.
OBSOLETE	VARCHAR2(3)	Indicates whether the backup piece or copy is obsolete (YES) or not (NO); valid only if FILE_TYPE is PIECE or COPY
BYTES	NUMBER	Size of the file (in bytes)
BS_KEY	NUMBER	Primary key of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_COUNT	NUMBER	Count of the backup set from the controlfile record (valid only if BACKUP_TYPE is BACKUP SET)
BS_STAMP	NUMBER	Stamp of the backup set from the controlfile record (valid only if BACKUP_TYPE is BACKUP SET)
BS_TYPE	VARCHAR2(32)	Type of the backup set (valid only if BACKUP_TYPE is BACKUP SET): <ul style="list-style-type: none"> ■ DATAFILE ■ ARCHIVED LOG
BS_INCR_TYPE	VARCHAR2(32)	Incremental type of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_PIECES	NUMBER	Number of backup pieces in the backup set (valid only if BACKUP_TYPE is BACKUP SET)

V\$BACKUP_PIECE

Column	Datatype	Description
BS_COPIES	NUMBER	Number of copies of the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BS_COMPLETION_TIME	DATE	Completion time of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_STATUS	VARCHAR2(16)	Status of the backup set (valid only if BACKUP_TYPE is BACKUP SET): <ul style="list-style-type: none">■ AVAILABLE■ UNAVAILABLE■ EXPIRED■ OTHER - Pieces which are part of the backup set do not have uniform status (that is, some of them are available, some not)
BS_BYTES	NUMBER	Sum of all backup piece sizes in the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_COMPRESSED	VARCHAR2(3)	Indicates whether the backup pieces of the backup set are compressed (YES) or not (NO); valid only if BACKUP_TYPE is BACKUP SET
BS_TAG	VARCHAR2(1024)	Tags of the backup set. If pieces have different tags, then all piece tags are concatenated and separated by commas. Valid only if BACKUP_TYPE is BACKUP SET)
BS_DEVICE_TYPE	VARCHAR2(255)	Device type of the backup set. If there is more than one device type, then they are separated by commas. Valid only if BACKUP_TYPE is BACKUP SET
BP_PIECE#	NUMBER	Number of pieces inside the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BP_COPY#	NUMBER	Number of copies of the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
DF_FILE#	NUMBER	Absolute file number of the datafile (valid only if FILE_TYPE is DATAFILE)
DF_TABLESPACE	VARCHAR2(30)	Tablespace name of the datafile (valid only if FILE_TYPE is DATAFILE)
DF_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the control file or datafile was created (valid only if FILE_TYPE is DATAFILE)
DF_CREATION_CHANGE#	NUMBER	Creation SCN of the control file or datafile (valid only if FILE_TYPE is DATAFILE)
DF_CHECKPOINT_CHANGE#	NUMBER	System change number (SCN) of the most recent control file or datafile checkpoint (valid only if FILE_TYPE is DATAFILE)
DF_CKP_MOD_TIME	DATE	Modification time in case of SPFILE, otherwise time when the control file or datafile was checkpointed (valid only if FILE_TYPE is DATAFILE)
RL_THREAD#	NUMBER	Redo log thread number of the archived log (valid only if FILE_TYPE is REDO LOG)
RL_SEQUENCE#	NUMBER	Redo log sequence number of the archived log (valid only if FILE_TYPE is REDO LOG)
RL_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the record was created (valid only if FILE_TYPE is REDO LOG)
RL_FIRST_CHANGE#	NUMBER	First SCN of the redo log (valid only if FILE_TYPE is REDO LOG)
RL_FIRST_TIME	DATE	Time when the Oracle Database switched into the redo log (valid only if FILE_TYPE is REDO LOG)
RL_NEXT_CHANGE#	NUMBER	First SCN of the next redo log in the thread (valid only if FILE_TYPE is REDO LOG)
RL_NEXT_TIME	DATE	First timestamp of the next redo log in the thread (valid only if FILE_TYPE is REDO LOG)

V\$BACKUP_PIECE

V\$BACKUP_PIECE displays information about backup pieces from the control file. Each backup set consists of one or more backup pieces.

Column	Datatype	Description
RECID	NUMBER	Backup piece record ID
STAMP	NUMBER	Backup piece record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	Backup piece number (1-N)
COPY#	NUMBER	Indicates the copy number for backup pieces created with duplex enabled. 1 if the backup piece is not duplexed.
DEVICE_TYPE	VARCHAR2 (17)	Type of the device on which the backup piece resides. Set to DISK for backup sets on disk. See Also: V\$BACKUP_DEVICE
HANDLE	VARCHAR2 (513)	Backup piece handle identifies the backup piece on restore
COMMENTS	VARCHAR2 (64)	Comment returned by the operating system or storage subsystem. Set to NULL for backup pieces on disk. This value is informational only; not needed for restore.
MEDIA	VARCHAR2 (65)	Name of the media on which the backup piece resides. This value is informational only; not needed for restore.
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager backup command.
CONCUR	VARCHAR2 (3)	(YES NO) Indicates whether the piece on a media that can be accessed concurrently
TAG	VARCHAR2 (32)	Backup piece tag. The tag is specified at backup set level, but stored at piece level.
STATUS	VARCHAR2 (1)	Indicates the status of the piece: A (available), D (deleted), or X (expired)
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
DELETED	VARCHAR2 (3)	(YES/NO) NO indicates that the file still exists. YES indicates the file no longer exists because it has been deleted.
BYTES	NUMBER	Size of the backup piece (in bytes)
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp
COMPRESSED	VARCHAR2 (3)	Indicates whether the backup piece is compressed (YES) or not (NO)

V\$BACKUP_REDOLOG

V\$BACKUP_REDOLOG displays information about archived logs in backup sets from the control file. Note that online redo logs cannot be backed up directly; they must be archived first to disk and then backed up. An archive log backup set can contain one or more archived logs.

Column	Datatype	Description
RECID	NUMBER	Record ID for this row; it is an integer that identifies this row
STAMP	NUMBER	Timestamp used with RECID to uniquely identify this row
SET_STAMP	NUMBER	One of the foreign keys for the row of the V\$BACKUP_SET table that identifies this backup set
SET_COUNT	NUMBER	One of the foreign keys for the row of the V\$BACKUP_SET table that identifies this backup set

V\$BACKUP_SET

Column	Datatype	Description
THREAD#	NUMBER	Thread number for the log
SEQUENCE#	NUMBER	Log sequence number
RESETLOGS_CHANGE#	NUMBER	Change number of the last resetlogs before the log was written
RESETLOGS_TIME	DATE	Change time of the last resetlogs before the log was written. These will be the same for all logs in a backup set.
FIRST_CHANGE#	NUMBER	SCN when the log was switched into. The redo in the log is at this SCN and greater.
FIRST_TIME	DATE	Time allocated when the log was switched into
NEXT_CHANGE#	NUMBER	SCN when the next log in this thread was switched into. The redo in the log is below this SCN.
NEXT_TIME	DATE	Time when the next log in this thread was switched into
BLOCKS	NUMBER	Size of the log in logical blocks including the header block
BLOCK_SIZE	NUMBER	Size of the log blocks in bytes

V\$BACKUP_SET

V\$BACKUP_SET displays information about backup sets from the control file. A backup set record is inserted after the backup set is successfully completed.

Column	Datatype	Description
RECID	NUMBER	Backup set record ID
STAMP	NUMBER	Backup set record stamp
SET_STAMP	NUMBER	Backup set stamp. The backup set stamp and count uniquely identify the backup set. Primary key for the V\$BACKUP_SET table, and the foreign key for the following tables: V\$BACKUP_PIECE, V\$BACKUP_DATAFILE, V\$BACKUP_REDOLOG, V\$BACKUP_CORRUPTION.
SET_COUNT	NUMBER	Backup set count. The backup set count is incremented by one every time a new backup set is started (if the backup set is never completed the number is "lost"). If the control file is re-created then the count is reset to 1. Therefore the count must be used with the stamp to uniquely identify a backup set. Primary key for the V\$BACKUP_SET table, and the foreign key for the following tables: V\$BACKUP_PIECE, V\$BACKUP_DATAFILE, V\$BACKUP_REDOLOG, V\$BACKUP_CORRUPTION
BACKUP_TYPE	VARCHAR2(1)	Type of files that are in this backup. If the backup contains archived redo logs, the value is L. If this is a datafile full backup, the value is D. If this is an incremental backup, the value is I.
CONTROLFILE_INCLUDED	VARCHAR2(3)	Set to YES if there is a control file included in this backup set, otherwise set to NO
INCREMENTAL_LEVEL	NUMBER	Location where this backup set fits into the database's backup strategy. Set to zero for full datafile backups, nonzero for incremental datafile backups, and NULL for archive log backups.
PIECES	NUMBER	Number of distinct backup pieces in the backup set
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Time that this backup set completed
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds
BLOCK_SIZE	NUMBER	Block size of the backup set
INPUT_FILE_SCAN_ONLY	VARCHAR2(3)	YES indicates no actual backup is performed, but the datafiles are read. NO indicates a normal backup is performed.
KEEP	VARCHAR2(3)	(YES/NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy

Column	Datatype	Description
KEEP_UNTIL	DATE	If KEEP_UNTIL_TIME is specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(10)	Lists additional retention options for this backup set. Possible values are: LOGS - The logs need to recover this backup are kept NOLOGS - The logs needed to recover this backup will not be kept

V\$BACKUP_SPFILE

V\$BACKUP_SPFILE displays information about server parameter files in backup sets from the control file.

Column	Datatype	Description
RECID	NUMBER	Backup SPFILE record ID
STAMP	NUMBER	Backup SPFILE record stamp
SET_STAMP	NUMBER	Backup set stamp (of the set which contains this SPFILE backup)
SET_COUNT	NUMBER	Backup set count (of the set which contains this SPFILE backup)
MODIFICATION_TIME	DATE	Time when the SPFILE was last modified (this also includes creation time)
BYTES	NUMBER	Size of the SPFILE (in bytes)
COMPLETION_TIME	DATE	Time when the backup of the SPFILE completed

V\$BACKUP_SYNC_IO

V\$BACKUP_SYNC_IO displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

See Also: *Oracle Database Performance Tuning Guide* for information on how to use this table to tune backup performance

Column	Datatype	Description
SID	NUMBER	The Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	The use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp
DEVICE_TYPE	VARCHAR2(17)	The device type where the file is located
TYPE	VARCHAR2(9)	INPUT, OUTPUT, or AGGREGATE
STATUS	VARCHAR2(11)	NOT_STARTED, IN_PROGRESS, or FINISHED
FILENAME	VARCHAR2(513)	The name of the backup file being read or written
SET_COUNT	NUMBER	The set count of the backup set being read or written
SET_STAMP	NUMBER	The set stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	The size of the buffers being used to read/write this file, in bytes
BUFFER_COUNT	NUMBER	The number of buffers being used to read/write this file
TOTAL_BYTES	NUMBER	The total number of bytes that will be read or written for this file, if known. If not known, this column will be null.

V\$BGPROCESS

Column	Datatype	Description
OPEN_TIME	DATE	The time this file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened.
CLOSE_TIME	DATE	The time this file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed.
ELAPSED_TIME	NUMBER	The time, in hundredths of a second, that the file was open
MAXOPENFILES	NUMBER	The number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'.
BYTES	NUMBER	The number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	The I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	The number of I/Os that were performed to this file
IO_TIME_TOTAL	NUMBER	The total time, in hundredths of a second, taken to do I/O for this file
IO_TIME_MAX	NUMBER	The maximum time taken for a single I/O request
DISCRETE_BYTES_PER_SECOND	NUMBER	The average transfer rate for this file

V\$BGPROCESS

V\$BGPROCESS displays information about the background processes.

Column	Datatype	Description
PADDR	RAW(4 8)	Address of the process state object
P SERIAL#	NUMBER	Process state object serial number
NAME	VARCHAR2(5)	Name of this background process
DESCRIPTION	VARCHAR2(64)	Description of the background process
ERROR	NUMBER	Error encountered

V\$BH

This is a Real Application Clusters view. This view gives the status and number of pings for every buffer in the SGA.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V\$DBFILE)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(6)	Status of the buffer: <ul style="list-style-type: none">free - Not currently in usexcur - Exclusivescur - Shared currentcr - Consistent readread - Being read from diskmrec - In media recovery modeirec - In instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.

Column	Datatype	Description
LOCK_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_CLASS	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
DIRTY	VARCHAR2(1)	Y - block modified
TEMP	VARCHAR2(1)	Y - temporary block
PING	VARCHAR2(1)	Y - block pinged
STALE	VARCHAR2(1)	Y - block is stale
DIRECT	VARCHAR2(1)	Y - direct block
NEW	VARCHAR2(1)	Always set to N. This column is obsolete and maintained for backward compatibility.
OBJD	NUMBER	Database object number of the block that the buffer represents
TS#	NUMBER	Tablespace number of block

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

V\$BLOCK_CHANGE_TRACKING

V\$BLOCK_CHANGE_TRACKING displays the status of block change tracking for the database.

Column	Datatype	Description
STATUS	VARCHAR2(10)	Status of block change tracking in the database: <ul style="list-style-type: none"> ■ DISABLED - Block change tracking is disabled ■ TRANSITION - Block change tracking is in the process of transitioning between the enabled and disabled states. The TRANSITION state should usually never be observed, because it only exists while enabling or disabling block change tracking. This state might be observed if the instance crashed while enabling or disabling block change tracking, in which case it will be cleaned up automatically the next time that the database is opened. ■ ENABLED - Block change tracking is enabled
FILENAME	VARCHAR2(513)	Name of the block change tracking file for the database
BYTES	NUMBER	Size of the block change tracking file (in bytes)

See Also: *Oracle Database Backup and Recovery Basics* for information on setting up block change tracking

V\$BUFFER_POOL

V\$BUFFER_POOL displays information about all buffer pools available for the instance.

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier number
NAME	VARCHAR2(20)	Name of the buffer pool: <ul style="list-style-type: none"> ■ DEFAULT ■ KEEP ■ RECYCLE <p>Note: Currently, KEEP and RECYCLE pools only exist for the standard block size. All non-standard block size pools are DEFAULT.</p>
BLOCK_SIZE	NUMBER	Block size (in bytes) for buffers in this pool. Possible values: the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, 32768.
RESIZE_STATE	VARCHAR2(10)	Current state of the resize operation: <ul style="list-style-type: none"> STATIC - Not being resized ALLOCATING - Memory is being allocated (can be cancelled by the user) ACTIVATING - New buffers are being created (user cannot cancel) SHRINKING - Buffers are being deleted (can be cancelled by the user)
CURRENT_SIZE	NUMBER	Present size of the sub-cache (in megabytes)
BUFFERS	NUMBER	Current instantaneous number of buffers
TARGET_SIZE	NUMBER	If a resize is in progress (state is not STATIC), records new target size (in megabytes). If the pool is STATIC, the value in this column is the same as the current size of the pool.
TARGET_BUFFERS	NUMBER	If a resize is in progress, records new target size in terms of buffers. Otherwise, the value in this column is the same as the current number of buffers.
PREV_SIZE	NUMBER	Previous buffer pool size. If the buffer pool has never been resized, the previous size is zero.
PREV_BUFFERS	NUMBER	Previous number of buffers in the buffer pool. Value is zero if the buffer pool has never been resized.
LO_BNUM	NUMBER	Obsolete column
HI_BNUM	NUMBER	Obsolete column
LO_SETID	NUMBER	Obsolete column
HI_SETID	NUMBER	Obsolete column
SET_COUNT	NUMBER	Obsolete column

See Also: ["DB_BLOCK_SIZE"](#) on page 1-35

V\$BUFFER_POOL_STATISTICS

V\$BUFFER_POOL_STATISTICS displays statistics about all buffer pools available for the instance.

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier number
NAME	VARCHAR2(20)	Name of the buffer pool
SET_MSIZ	NUMBER	Buffer pool maximum set size
CNUM_REPL	NUMBER	Number of buffers on replacement list

Column	Datatype	Description
CNUM_WRITE	NUMBER	Number of buffers on write list
CNUM_SET	NUMBER	Number of buffers in set
BUF_GOT	NUMBER	Number of buffers gotten by the set
SUM_WRITE	NUMBER	Number of buffers written by the set
SUM_SCAN	NUMBER	Number of buffers scanned in the set
FREE_BUFFER_WAIT	NUMBER	Free buffer wait statistic
WRITE_COMPLETE_WAIT	NUMBER	Write complete wait statistic
BUFFER_BUSY_WAIT	NUMBER	Buffer busy wait statistic
FREE_BUFFER_INSPECTED	NUMBER	Free buffer inspected statistic
DIRTY_BUFFERS_INSPECTED	NUMBER	Dirty buffers inspected statistic
DB_BLOCK_CHANGE	NUMBER	Database blocks changed statistic
DB_BLOCK_GETS	NUMBER	Database blocks gotten statistic
CONSISTENT_GETS	NUMBER	Consistent gets statistic
PHYSICAL_READS	NUMBER	Physical reads statistic
PHYSICAL_WRITES	NUMBER	Physical writes statistic

See Also: "DB_CACHE_SIZE" on page 1-36

V\$BUFFERED_PUBLISHERS

V\$BUFFERED_PUBLISHERS displays information about all buffered publishers in the instance. There is one row per queue per sender. The values are reset to zero when the database (or instance in a RAC environment) restarts.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
SENDER_NAME	VARCHAR2(30)	Name of the agent enqueueing the message (the Streams name assigned for a capture process)
SENDER_ADDRESS	VARCHAR2(1024)	Queue name and database name of the source (last propagating) queue; database name is not specified if the source queue is in the local database
SENDER_PROTOCOL	NUMBER	Protocol used by the sender's address
NUM_MSGS	NUMBER	Current number of messages that have yet to be dequeued from the buffer queue
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued into the buffered queue since the buffered queue was created (comparable to V\$STREAMS_CAPTURE total_enqueued_messages)
LAST_ENQUEUED_MSG	NUMBER	Most recently enqueued message identifier

V\$BUFFERED_QUEUES

V\$BUFFERED_QUEUES displays information about all buffered queues in the instance. There is one row per queue.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue

V\$BUFFERED_SUBSCRIBERS

Column	Datatype	Description
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
STARTUP_TIME	DATE	Startup time
NUM_MSGS	NUMBER	Total number of messages currently in the buffered queue
SPILL_MSGS	NUMBER	Current number of overflow messages spilled to disk from the buffered queue
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued into the buffered since the buffered queue was created
CSPILL_MSGS	NUMBER	Cumulative total number of overflow messages spilled to disk from the buffered queue since the buffered queue was created

V\$BUFFERED_SUBSCRIBERS

V\$BUFFERED_SUBSCRIBERS displays information about the subscribers for all buffered queues in the instance. There is one row per subscriber per queue.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Schema of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
SUBSCRIBER_ID	NUMBER	Internal subscriber number (for identification)
SUBSCRIBER_NAME	VARCHAR2(30)	Name of the subscriber (comparable to the Streams name for the Streams process)
SUBSCRIBER_ADDRESS	VARCHAR2(1024)	Address of the subscribing agent
PROTOCOL	NUMBER	Protocol of the subscribing agent
SUBSCRIBER_TYPE	VARCHAR2(30)	Type of the subscriber: <ul style="list-style-type: none">■ PROXY - Proxy subscriber■ SUBSCRIBER
STARTUP_TIME	DATE	Startup time
LAST_BROWSED_SEQ	NUMBER	Sequence number of the most recently browsed message for the subscriber (comparable to the number of messages in the V\$STREAMS_APPLY_READER view)
LAST_BROWSED_NUM	NUMBER	Internal Message number for the most recently browsed message for the subscriber
LAST_DEQUEUED_SEQ	NUMBER	Sequence number of the most recently dequeued message for the subscriber (comparable to the number of messages in the V\$STREAMS_APPLY_COORDINATOR view)
LAST_DEQUEUED_NUM	NUMBER	Internal Message number for the most recently dequeued message for the subscriber
CURRENT_ENQ_SEQ	NUMBER	Current sequence number of the most recently enqueued message for the subscriber
NUM_MSGS	NUMBER	Total number of outstanding messages currently enqueued in the buffered queue for the subscriber (includes the count of the messages overflowed to disk)
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued for the subscriber since the creation of the buffered queue
TOTAL_DEQUEUED_MSG	NUMBER	Total number of messages dequeued by the subscriber
TOTAL_SPILLED_MSG	NUMBER	Total number of spilled messages for the subscriber

V\$CACHE

This is a Real Application Clusters view. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V\$DBFILE)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(6)	Status of the block: <ul style="list-style-type: none"> ■ free - Not currently in use ■ xcur - Exclusive ■ scur - Shared current ■ cr - Consistent read ■ read - Being read from disk ■ mrec - In media recovery mode ■ irec - In instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	Name of the partition (null for non-partitioned objects)

Column	Datatype	Description
KIND	VARCHAR2(15)	Type of the database object: <ul style="list-style-type: none"> ▪ INDEX ▪ TABLE ▪ CLUSTER ▪ VIEW ▪ SYNONYM ▪ SEQUENCE ▪ PROCEDURE ▪ FUNCTION ▪ PACKAGE ▪ NON-EXISTENT ▪ PACKAGE BODY ▪ TRIGGER ▪ TYPE ▪ TYPE BODY ▪ TABLE PARTITION ▪ INDEX PARTITION ▪ LOB ▪ LIBRARY ▪ JAVA SOURCE ▪ JAVA CLASS ▪ JAVA RESOURCE ▪ JAVA DATA ▪ UNDO
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Name of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

V\$CACHE_LOCK

This is a Real Application Clusters view. V\$CACHE_LOCK is similar to V\$CACHE, except for the platform specific lock manager identifiers. This information may be useful if the platform-specific lock manager provides tools for monitoring the PCM lock operations that are occurring. For example, first query to find the lock element address using INDX and CLASS, then query V\$BH to find the buffers that are covered by the lock.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V\$DBFILE)
BLOCK#	NUMBER	Block number

Column	Datatype	Description
STATUS	VARCHAR2 (6)	Status of the block: <ul style="list-style-type: none"> ▪ free - Not currently in use ▪ xcur - Exclusive ▪ scur - Shared current ▪ cr - Consistent read ▪ read - Being read from disk ▪ mrec - In media recovery mode ▪ irec - In instance recovery mode
XNC	NUMBER	Number of parallel cache management (PCM) lock conversions due to contention with another instance
NAME	VARCHAR2 (30)	Name of the database object containing the block
KIND	VARCHAR2 (12)	Type of database object: <ul style="list-style-type: none"> ▪ 1 - Index ▪ 2 - Table ▪ 3 - Cluster ▪ 4 - View ▪ 5 - Synonym ▪ 6 - Sequence ▪ 7 - Procedure ▪ 8 - Function ▪ 9 - Package ▪ 10 - Nonexistent ▪ 11 - Package body ▪ 12 - Trigger ▪ 13 - Type ▪ 14 - Type body ▪ 19 - Table partition ▪ 20 - Index partition ▪ 21 - LOB ▪ 22 - Library ▪ Null - Unknown
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW (4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
INDX	NUMBER	Platform-specific lock manager identifier
CLASS	NUMBER	Platform-specific lock manager identifier

See Also:

- *Oracle Real Application Clusters Installation and Configuration Guide*
- "[V\\$CACHE](#)" on page 4-31

V\$CACHE_TRANSFER

This is a Real Application Clusters view. The V\$CACHE_TRANSFER view is identical to the V\$CACHE view but only displays blocks that have been pinged at least once. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V\$DBFILE)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(6)	Status of the block: <ul style="list-style-type: none"> ▪ free - Not currently in use ▪ xcur - Exclusive ▪ scur - Shared current ▪ cr - Consistent read ▪ read - Being read from disk ▪ mrec - In media recovery mode ▪ irec - In instance recovery mode
XNC	NUMBER	Number of PCM lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	NULL for non-partitioned objects
KIND	VARCHAR2(15)	Type of database object See Also: Table 4-1 on page 4-77
OWNER#	NUMBER	Owner number
GC_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
GC_ELEMENT_NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer

See Also: "[V\\$CACHE](#)" on page 4-31 and *Oracle Real Application Clusters Installation and Configuration Guide*

V\$CIRCUIT

V\$CIRCUIT contains information about virtual circuits, which are user connections to the database through dispatchers and servers.

Column	Datatype	Description
CIRCUIT	RAW(4 8)	Circuit address
DISPATCHER	RAW(4 8)	Current dispatcher process address
SERVER	RAW(4 8)	Current server process address
WAITER	RAW(4 8)	Address of the server process that is waiting for the (currently busy) circuit to become available
SADDR	RAW(4 8)	Address of the session bound to the circuit
STATUS	VARCHAR2(16)	Status of the circuit: <ul style="list-style-type: none"> ■ BREAK - currently interrupted ■ EOF - about to be removed ■ OUTBOUND - an outward link to a remote database ■ NORMAL - normal circuit into the local database
QUEUE	VARCHAR2(16)	Queue the circuit is currently on: <ul style="list-style-type: none"> ■ COMMON - on the common queue, waiting to be picked up by a server process ■ DISPATCHER - waiting for the dispatcher ■ SERVER - currently being serviced ■ NONE - idle circuit
MESSAGE0	NUMBER	Size in bytes of the messages in the first message buffer
MESSAGE1	NUMBER	Size in bytes of the messages in the second message buffer
MESSAGE2	NUMBER	Size in bytes of the messages in the third message buffer
MESSAGE3	NUMBER	Size in bytes of the messages in the fourth message buffer
MESSAGES	NUMBER	Total number of messages that have gone through this circuit
BYTES	NUMBER	Total number of bytes that have gone through this circuit
BREAKS	NUMBER	Total number of breaks (interruptions) for this circuit
PRESENTATION	VARCHAR2(257)	Presentation protocol used by the client and server
PCIRCUIT	RAW(4 8)	Address of the parent circuit

V\$CLASS_CACHE_TRANSFER

V\$CLASS_CACHE_TRANSFER is deprecated.

Column	Datatype	Description
CLASS	CHAR(10)	Block class; always data block
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0
CR_TRANSFER	NUMBER	Number of CR blocks transferred; always 0
CURRENT_TRANSFER	NUMBER	Number of current blocks transferred; always 0

V\$CLIENT_STATS

V\$CLIENT_STATS displays measures for all sessions that are active for the client identifier per instance. The statistics available in this view are a subset of those available in V\$SESSTAT and V\$SESS_TIME_MODEL%.

Column	Datatype	Description
CLIENT_IDENTIFIER	VARCHAR2(64)	Client identifier
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2(64)	Derived statistic name from V\$STATNAME and V\$SESS_TIME_MODEL%
VALUE	NUMBER	Cumulative value

V\$CONTEXT

This view lists set attributes in the current session.

Column	Datatype	Description
NAMESPACE	VARCHAR2(31)	Namespace that the attribute is in
ATTRIBUTE	VARCHAR2(31)	Name of the attribute
VALUE	VARCHAR2(4000)	Value of the attribute

V\$CONTROLFILE

This view lists the names of the control files.

Column	Datatype	Description
STATUS	VARCHAR2(7)	INVALID if the name cannot be determined, which should not occur. Null if the name can be determined.
NAME	VARCHAR2(513)	Name of the control file
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)

V\$CONTROLFILE_RECORD_SECTION

V\$CONTROLFILE_RECORD_SECTION displays information about the control file record sections.

Column	Datatype	Description
TYPE	VARCHAR2 (28)	Identifies the type of record section: <ul style="list-style-type: none"> ■ DATABASE ■ CKPT PROGRESS ■ REDO THREAD ■ REDO LOG ■ DATAFILE ■ FILENAME ■ TABLESPACE ■ TEMPORARY FILENAME ■ RMAN CONFIGURATION ■ LOG HISTORY ■ OFFLINE RANGE ■ ARCHIVED LOG ■ BACKUP SET ■ BACKUP PIECE ■ BACKUP DATAFILE ■ BACKUP REDOLOG ■ DATAFILE COPY ■ BACKUP CORRUPTION ■ COPY CORRUPTION ■ DELETED OBJECT ■ PROXY COPY ■ BACKUP SPFILE ■ DATABASE INCARNATION ■ FLASHBACK LOG ■ RECOVERY DESTINATION ■ INSTANCE SPACE RESERVATION ■ REMOVABLE RECOVERY FILES ■ RMAN STATUS ■ THREAD INSTANCE NAME MAPPING ■ MTR ■ DATAFILE HISTORY
RECORD_SIZE	NUMBER	Record size in bytes
RECORDS_TOTAL	NUMBER	Number of records allocated for the section
RECORDS_USED	NUMBER	Number of records used in the section
FIRST_INDEX	NUMBER	Index (position) of the first record
LAST_INDEX	NUMBER	Index of the last record
LAST_RECID	NUMBER	Record ID of the last record

V\$COPY_CORRUPTION

This view displays information about datafile copy corruptions from the control file.

Column	Datatype	Description
RECID	NUMBER	Copy corruption record ID
STAMP	NUMBER	Copy corruption record stamp
COPY_RECID	NUMBER	Datafile copy record ID
COPY_STAMP	NUMBER	Datafile copy record stamp

Column	Datatype	Description
FILE#	NUMBER	Datafile number
BLOCK#	NUMBER	First block of the corrupted range
BLOCKS	NUMBER	Number of contiguous blocks in the corrupted range
CORRUPTION_CHANGE#	NUMBER	Change# at which the logical corruption was detected. Set to 0 to indicate media corruption.
MARKED_CORRUPT	VARCHAR2(3)	(YES NO) If set to YES the blocks were not marked corrupted in the datafile, but were detected and marked as corrupted while making the datafile copy
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: <ul style="list-style-type: none"> ■ ALL_ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ■ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ■ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ■ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ■ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V\$CR_BLOCK_SERVER

This view displays statistics on the Global Cache Service processes (LMS) used in cache fusion.

Column	Datatype	Description
CR_REQUESTS	NUMBER	Number of CR blocks served due to remote CR block requests
CURRENT_REQUESTS	NUMBER	Number of current blocks served due to remote CR block requests CR_REQUESTS + CURRENT_REQUESTS = global cache cr clocks served (from V\$SYSSTAT).
DATA_REQUESTS	NUMBER	Number of current or CR requests for data blocks
UNDO_REQUESTS	NUMBER	Number of CR requests for undo blocks
TX_REQUESTS	NUMBER	Number of CR requests for undo segment header blocks DATA_REQUESTS + UNDO_REQUESTS + TX_REQUESTS = total number of requests handled by the LMS processes
CURRENT_RESULTS	NUMBER	Number of requests for which no changes were rolled out of the block returned to the requesting instance
PRIVATE_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance, and only the requesting transaction can use the resulting CR block
ZERO_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance. Only zero-XID transactions can use the block.
DISK_READ_RESULTS	NUMBER	Number of requests for which the requesting instance had to read the requested block from disk
FAIL_RESULTS	NUMBER	Number of requests that failed; the requesting transaction must reissue the request
FAIRNESS_DOWN_CONVERTS	NUMBER	Number of times an instance receiving a request has down-converted an X lock on a block because it was not modifying the block

Column	Datatype	Description
FAIRNESS_CLEARS	NUMBER	Number of times the "fairness counter" was cleared. This counter tracks the number of times a block was modified after it was served.
FREE_GC_ELEMENTS	NUMBER	Number of times a request was received from another instance and the X lock had no buffers
FLUSHES	NUMBER	Number of times the log has been flushed by an LMS process
FLUSHES_QUEUED	NUMBER	Number of flushes queued by an LMS process
FLUSH_QUEUE_FULL	NUMBER	Number of times the flush queue was full
FLUSH_MAX_TIME	NUMBER	Maximum time for flush
LIGHT_WORKS	NUMBER	Number of times the light-work rule was evoked. This rule prevents the LMS processes from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the LMS process from completing its response to the CR request.
ERRORS	NUMBER	Number of times an error was signalled by an LMS process

See Also: *Oracle Real Application Clusters Installation and Configuration Guide* for more information on these processes and on cache fusion

Note: This view contains internal diagnostic information for use by Oracle Support Services. It is subject to change without notice.

V\$CURRENT_BLOCK_SERVER

V\$CURRENT_BLOCK_SERVER displays statistics on the Global Cache Service processes (IMS) used in cache fusion.

Column	Datatype	Description
PIN1	NUMBER	Pins taking less than 1 millisecond
PIN10	NUMBER	Pins taking 1 to 10 milliseconds
PIN100	NUMBER	Pins taking 10 to 100 milliseconds
PIN1000	NUMBER	Pins taking 100 to 1000 milliseconds
PIN10000	NUMBER	Pins taking 1000 to 10000 milliseconds
FLUSH1	NUMBER	Flushes taking less than 1 millisecond
FLUSH10	NUMBER	Flushes taking 1 to 10 milliseconds
FLUSH100	NUMBER	Flushes taking 10 to 100 milliseconds
FLUSH1000	NUMBER	Flushes taking 100 to 1000 milliseconds
FLUSH10000	NUMBER	Flushes taking 1000 to 10000 milliseconds
WRITE1	NUMBER	Writes taking less than 1 millisecond
WRITE10	NUMBER	Writes taking 1 to 10 milliseconds
WRITE100	NUMBER	Writes taking 10 to 100 milliseconds
WRITE1000	NUMBER	Writes taking 100 to 1000 milliseconds
WRITE10000	NUMBER	Writes taking 1000 to 10000 milliseconds

V\$DATABASE

V\$DATABASE displays information about the database from the control file.

Column	Datatype	Description
DBID	NUMBER	Database identifier calculated when the database is created and stored in all file headers
NAME	VARCHAR2(9)	Name of the database
CREATED	DATE	Creation date of the database
RESETLOGS_CHANGE#	NUMBER	System change number (SCN) at open resetlogs
RESETLOGS_TIME	DATE	Timestamp of open resetlogs
PRIOR_RESETLOGS_CHANGE#	NUMBER	SCN at prior resetlogs
PRIOR_RESETLOGS_TIME	DATE	Timestamp of prior resetlogs
LOG_MODE	VARCHAR2(12)	Archive log mode: <ul style="list-style-type: none"> ■ NOARCHIVELOG ■ ARCHIVELOG ■ MANUAL
CHECKPOINT_CHANGE#	NUMBER	Last SCN checkpointed
ARCHIVE_CHANGE#	NUMBER	Database force archiving SCN. Any redo log with a start SCN below this will be forced to archive out.
CONTROLFILE_TYPE	VARCHAR2(7)	Type of control file: <ul style="list-style-type: none"> ■ STANDBY - Indicates that the database is in standby mode ■ CLONE - Indicates a clone database ■ BACKUP CREATED - Indicates the database is being recovered using a backup or created control file ■ CURRENT - database is available for general use
CONTROLFILE_CREATED	DATE	Creation date of the control file
CONTROLFILE_SEQUENCE#	NUMBER	Control file sequence number incremented by control file transactions
CONTROLFILE_CHANGE#	NUMBER	Last SCN in backup control file; null if the control file is not a backup
CONTROLFILE_TIME	DATE	Last timestamp in backup control file; null if the control file is not a backup
OPEN_RESETLOGS	VARCHAR2(11)	(NOT ALLOWED ALLOWED REQUIRED) Indicates whether the next database open allows or requires the resetlogs option
VERSION_TIME	DATE	Version time
OPEN_MODE	VARCHAR2(10)	Open mode information: <ul style="list-style-type: none"> ■ MOUNTED ■ READ WRITE ■ READ ONLY
PROTECTION_MODE	VARCHAR2(20)	Protection mode currently in effect for the database: <ul style="list-style-type: none"> ■ MAXIMUM PROTECTION - Database is running in maximized protection mode ■ MAXIMUM AVAILABILITY - Database is running in maximized availability mode ■ RESYNCHRONIZATION - Database is running in resynchronization mode ■ MAXIMUM PERFORMANCE - Database is running in maximized protection mode ■ UNPROTECTED - Database is unprotected (this normally occurs when the primary database is mounted and not open)

Column	Datatype	Description
PROTECTION_LEVEL	VARCHAR2(20)	<p>Aggregated protection mode currently in effect for the database:</p> <ul style="list-style-type: none"> ■ MAXIMUM PROTECTION - Database is running in maximized protection mode ■ MAXIMUM AVAILABILITY - Database is running in maximized availability mode ■ RESYNCHRONIZATION - Database is running in resynchronization mode ■ MAXIMUM PERFORMANCE - Database is running in maximized protection mode ■ UNPROTECTED - Database is unprotected (this normally occurs when the primary database is mounted and not open) <p>Note:This column is an aggregation of the PROTECTION_MODE of all standby archive log destinations.</p>
REMOTE_ARCHIVE	VARCHAR2(8)	Value of the REMOTE_ARCHIVE_ENABLE initialization parameter
ACTIVATION#	NUMBER	Number assigned to the database instantiation
SWITCHOVER#	NUMBER	Number assigned to the database switchover
DATABASE_ROLE	VARCHAR2(16)	<p>Current role of the database:</p> <ul style="list-style-type: none"> ■ LOGICAL STANDBY ■ PHYSICAL STANDBY ■ PRIMARY
ARCHIVELOG_CHANGE#	NUMBER	Highest NEXT_CHANGE# (from the V\$ARCHIVED_LOG view) for an archive log
SWITCHOVER_STATUS	VARCHAR2(20)	<p>Indicates whether switchover is allowed:</p> <ul style="list-style-type: none"> ■ NOT ALLOWED - Either this is a standby database and the primary database has not been switched first or this is a primary database and there are no standby databases. ■ SESSIONS ACTIVE - Indicates that there are active SQL sessions attached to the primary or standby database that need to be disconnected before the switchover operation is permitted. Query the V\$SESSION view to identify the specific processes that need to be terminated. ■ SWITCHOVER PENDING - This is a standby database and the primary database switchover request has been received but not processed. ■ SWITCHOVER LATENT - The switchover was in pending mode, but did not complete and went back to the primary database. ■ TO PRIMARY - This is a standby database and is allowed to switch over to a primary database. ■ TO STANDBY - This is a primary database and is allowed to switch over to a standby database. ■ RECOVERY NEEDED - This is a standby database that has not received the switchover request. ■ PREPARING SWITCHOVER - Either this is a primary database that is accepting redo data from a logical standby database in preparation for switch over to the logical standby database role, or it is a logical standby database sending redo data to a primary database and other standby databases in preparation for switch over to the primary database role. In the latter case, a completed dictionary has already been sent to the primary database and other standby databases. ■ PREPARING DICTIONARY - This is a logical standby database that is sending redo data to a primary database and other standby databases in the configuration in preparation for switch over to the primary database role. ■ TO LOGICAL STANDBY - This is a primary database that has received a complete dictionary from a logical standby database.
DATAGUARD_BROKER	VARCHAR2(8)	Indicates whether the Data Guard configuration is being managed by the broker (ENABLED) or not (DISABLED)

Column	Datatype	Description
GUARD_STATUS	VARCHAR2(7)	Protects data from being changed: <ul style="list-style-type: none"> ALL - Indicates all users other than SYS are prevented from making changes to any data in the database. STANDBY - Indicates all users other than SYS are prevented from making changes to any database object being maintained by logical standby. NONE - Indicates normal security for all data in the database.
SUPPLEMENTAL_LOG_DATA_MIN	VARCHAR2(8)	Ensures that LogMiner (and any products building on LogMiner technology) will have sufficient information to support chained rows and various storage arrangements such as cluster tables: <ul style="list-style-type: none"> NO - None of the database-wide supplemental logging directives are enabled IMPLICIT - Minimal supplemental logging is enabled because all or a combination of primary key, unique key, and foreign key supplemental logging is enabled YES - Minimal supplemental logging is enabled through an ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement <p>See Also: <i>Oracle Database SQL Reference</i> for additional information about the ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement</p>
SUPPLEMENTAL_LOG_DATA_PK	VARCHAR2(3)	For all tables with a primary key, indicates whether all columns of the primary key are placed into the redo log whenever an update is performed (YES) or not (NO) <p>See Also: <i>Oracle Database SQL Reference</i> for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG <i>supplemental_id_key_clause</i> statement</p>
SUPPLEMENTAL_LOG_DATA_UI	VARCHAR2(3)	For all tables with a unique key, indicates whether all other columns belonging to the unique key are placed into the redo log if any of the unique key columns are modified (YES) or not (NO) <p>See Also: <i>Oracle Database SQL Reference</i> for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG <i>supplemental_id_key_clause</i> statement</p>
FORCE_LOGGING	VARCHAR2(3)	Indicates whether the database is under force logging mode (YES) or not (NO)
PLATFORM_ID	NUMBER	Platform identification number of the database
PLATFORM_NAME	VARCHAR2(101)	Platform name of the database
RECOVERY_TARGET_INCARNATION#	NUMBER	Incarnation number where all datafiles are recovered by the RECOVER DATABASE command
LAST_OPEN_INCARNATION#	NUMBER	Record number of the incarnation in V\$DATABASE_INCARNATION that was last opened successfully
CURRENT_SCN	NUMBER	Current SCN; null if the database is not currently open. For a standby database, the current standby redo application SCN.
FLASHBACK_ON	VARCHAR2(3)	Indicates whether the database is in FLASHBACK mode (YES) or not (NO)
SUPPLEMENTAL_LOG_DATA_FK	VARCHAR2(3)	For all tables with a foreign key, indicates whether all other columns belonging to the foreign key are placed into the redo log if any foreign key columns are modified (YES) or not (NO) <p>See Also: <i>Oracle Database SQL Reference</i> for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG <i>supplemental_id_key_clause</i> statement</p>
SUPPLEMENTAL_LOG_DATA_ALL	VARCHAR2(3)	For all columns, indicates whether all the fixed-length maximum size columns of that row are placed into the redo log (YES) or not (NO) <p>See Also: <i>Oracle Database SQL Reference</i> for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG <i>supplemental_id_key_clause</i> statement</p>
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name

Column	Datatype	Description
STANDBY_BECAME_PRIMARY_SCN	NUMBER	SCN at which a physical standby database became a primary database. This SCN is useful for converting a failed primary database into a physical standby database after a forced failover. <i>See Also: Oracle Data Guard Concepts and Administration.</i>

V\$DATABASE_BLOCK_CORRUPTION

V\$DATABASE_BLOCK_CORRUPTION displays information about database blocks that were corrupted after the last backup.

Column	Datatype	Description
FILE#	NUMBER	Absolute file number of the datafile that contains the corrupt blocks
BLOCK#	NUMBER	Block number of the first corrupt block in the range of corrupted blocks
BLOCKS	NUMBER	Number of corrupted blocks found starting with BLOCK#
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: <ul style="list-style-type: none"> ▪ ALL_ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ▪ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ▪ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ▪ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ▪ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V\$DATABASE_INCARNATION

V\$DATABASE_INCARNATION displays information about all database incarnations. Oracle creates a new incarnation whenever a database is opened with the RESETLOGS option. Records about the current and immediately previous incarnation are also contained in the V\$DATABASE view.

Column	Datatype	Description
INCARNATION#	NUMBER	Record ID for the branch record in the controlfile
RESETLOGS_CHANGE#	NUMBER	Resetlogs system change number (SCN) for the incarnation of the current row
RESETLOGS_TIME	DATE	Resetlogs timestamp for the incarnation of the current row
PRIOR_RESETLOGS_CHANGE#	NUMBER	Resetlogs SCN for the previous incarnation
PRIOR_RESETLOGS_TIME	DATE	Resetlogs timestamp for the previous incarnation
STATUS	VARCHAR2(7)	Incarnation status: <ul style="list-style-type: none"> ▪ ORPHAN - Orphan incarnation ▪ CURRENT - Current incarnation of the database ▪ PARENT - Parent of the current incarnation

Column	Datatype	Description
RESETLOGS_ID	NUMBER	Branch ID for the incarnation of the current row (used by user-managed recovery/RMAN restore to get unique names for archived logs across incarnations)
PRIOR_INCARNATION#	NUMBER	Parent incarnation record ID if nonzero

V\$DATAFILE

This view contains datafile information from the control file.

See Also: ["V\\$DATAFILE_HEADER"](#) on page 4-46, which displays information from datafile headers

Column	Datatype	Description
FILE#	NUMBER	File identification number
CREATION_CHANGE#	NUMBER	Change number at which the datafile was created
CREATION_TIME	DATE	Timestamp of the datafile creation
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
STATUS	VARCHAR2(7)	Type of file (system or user) and its status. Values: OFFLINE, ONLINE, SYSTEM, RECOVER, SYSOFF (an offline file from the SYSTEM tablespace)
ENABLED	VARCHAR2(10)	Describes how accessible the file is from SQL: <ul style="list-style-type: none"> ■ DISABLED - No SQL access allowed ■ READ ONLY - No SQL updates allowed ■ READ WRITE - Full access allowed ■ UNKNOWN - should not occur unless the control file is corrupted
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Timestamp of the checkpoint#
UNRECOVERABLE_CHANGE#	NUMBER	Last unrecoverable change# made to this datafile. This column is always updated when an unrecoverable operation completes.
UNRECOVERABLE_TIME	DATE	Timestamp of the last unrecoverable change
LAST_CHANGE#	NUMBER	Last change# made to this datafile. Set to NULL if the datafile is being changed.
LAST_TIME	DATE	Timestamp of the last change
OFFLINE_CHANGE#	NUMBER	Offline change# of the last offline range. This column is updated only when the datafile is brought online.
ONLINE_CHANGE#	NUMBER	Online change# of the last offline range
ONLINE_TIME	DATE	Online timestamp of the last offline range
BYTES	NUMBER	Current datafile size (in bytes); 0 if inaccessible
BLOCKS	NUMBER	Current datafile size (in blocks); 0 if inaccessible
CREATE_BYTES	NUMBER	Size when created (in bytes)
BLOCK_SIZE	NUMBER	Block size of the datafile
NAME	VARCHAR2(513)	Name of the datafile
PLUGGED_IN	NUMBER	Describes whether the tablespace is plugged in. The value is 1 if the tablespace is plugged in and has not been made read/write, 0 if not.
BLOCK1_OFFSET	NUMBER	Offset from the beginning of the file to where the Oracle generic information begins. The exact length of the file can be computed as follows: BYTES + BLOCK1_OFFSET.
AUX_NAME	VARCHAR2(513)	Auxiliary name that has been set for this file via CONFIGURE AUXNAME

V\$DATAFILE_COPY

This view displays datafile copy information from the control file.

Column	Datatype	Description
RECID	NUMBER	Datafile copy record ID
STAMP	NUMBER	Datafile copy record stamp
NAME	VARCHAR2 (513)	Filename of the datafile copy. The maximum length of the name is OS dependent.
TAG	VARCHAR2 (32)	Datafile copy tag
FILE#	NUMBER	Absolute datafile number
RFILE#	NUMBER	Tablespace relative datafile number
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made
INCREMENTAL_LEVEL	NUMBER	The incremental level
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change# of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change seen when the datafile was copied
RECOVERY_FUZZY_CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY_TIME	DATE	Timestamp of the highest change written to the file by media recovery
ONLINE_FUZZY	VARCHAR2 (3)	(YES NO) If set to YES, this is a copy taken using an operating system utility after a crash or offline immediate (or an invalid copy taken while datafile was online and the database open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent.
BACKUP_FUZZY	VARCHAR2 (3)	(YES NO) If set to YES, this is a copy taken using the BEGIN BACKUP / END BACKUP technique. Recovery will need to apply all redo up to the end backup marker to make this copy consistent.
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt by this copy operation. That is, blocks that were not marked corrupted in the source datafile, but were detected and marked as corrupted during the copy operation.
MEDIA_CORRUPT	NUMBER	Total number of media corrupt blocks. For example, blocks with checksum errors are marked media corrupt.
LOGICALLY_CORRUPT	NUMBER	Total number of logically corrupt blocks. For example, applying redo for unrecoverable operations will mark affected blocks logically corrupt.
BLOCKS	NUMBER	Size of the datafile copy in blocks (also the size of the datafile when the copy was made)
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE_RANGE	NUMBER	RECID of the oldest offline range record in this control file copy; 0 for datafile copies
DELETED	VARCHAR2 (3)	(YES NO) If set to YES the datafile copy has been deleted or overwritten
STATUS	VARCHAR2 (1)	Identifies the status of this datafile copy. Possible values are: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the copy was completed

V\$DATAFILE_HEADER

Column	Datatype	Description
CONTROLFILE_TYPE	VARCHAR2(1)	B indicates normal copies S indicates standby copies
KEEP	VARCHAR2(3)	(YES/NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy
KEEP_UNTIL	DATE	If KEEP_UNTIL_TIME is specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(10)	Lists additional retention options for this backup set. Possible values are: LOGS - The logs need to recover this backup are kept NOLOGS - The logs needed to recover this backup will not be kept
SCANNED	VARCHAR2(3)	Indicates whether RMAN scanned the file (YES) or not (NO)
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp

V\$DATAFILE_HEADER

This view displays datafile information from the datafile headers.

Column	Datatype	Description
FILE#	NUMBER	Datafile number (from control file)
STATUS	VARCHAR2(7)	ONLINE OFFLINE (from control file)
ERROR	VARCHAR2(18)	NULL if the datafile header read and validation were successful. If the read failed then the rest of the columns are NULL. If the validation failed then the rest of columns may display invalid data. If there is an error then usually the datafile must be restored from a backup before it can be recovered or used.
FORMAT	NUMBER	Indicates the format for the header block. The possible values are 6, 7, 8, or 0. 6 - indicates Oracle Version 6 7 - indicates Oracle Version 7 8 - indicates Oracle Version 8 0 - indicates the format could not be determined (for example, the header could not be read)
RECOVER	VARCHAR2(3)	File needs media recovery (YES NO)
FUZZY	VARCHAR2(3)	File is fuzzy (YES NO)
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
TABLESPACE_NAME	VARCHAR2(30)	Tablespace name
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change#
RESETLOGS_TIME	DATE	Resetlogs timestamp
CHECKPOINT_CHANGE#	NUMBER	Datafile checkpoint change#
CHECKPOINT_TIME	DATE	Datafile checkpoint timestamp
CHECKPOINT_COUNT	NUMBER	Datafile checkpoint count
BYTES	NUMBER	Current datafile size in bytes
BLOCKS	NUMBER	Current datafile size in blocks

Column	Datatype	Description
NAME	VARCHAR2(513)	Datafile name

V\$DATAGUARD_CONFIG

V\$DATAGUARD_CONFIG displays the unique database names defined with the DB_UNIQUE_NAME and LOG_ARCHIVE_CONFIG initialization parameters, providing a view of the Data Guard environment from any database in the configuration.

The first row of the view lists the unique database name of the current database that was specified with the DB_UNIQUE_NAME initialization parameter. Additional rows reflect the unique database names of the other databases in the configuration that were specified with the DG_CONFIG keyword of the LOG_ARCHIVE_CONFIG initialization parameter.

Column	Datatype	Description
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name

V\$DATAGUARD_STATUS

V\$DATAGUARD_STATUS displays and logs events that would typically be triggered by any message to the alert log or server process trace files.

Column	Datatype	Description
FACILITY	VARCHAR2(24)	Facility that encountered the event: <ul style="list-style-type: none"> ▪ Crash Recovery ▪ Log Transport Services ▪ Log Apply Services ▪ Role Management Services ▪ Remote File Server ▪ Fetch Archive Log ▪ Data Guard ▪ Network Services
SEVERITY	VARCHAR2(13)	Severity of the event: <ul style="list-style-type: none"> ▪ Informational - Informational message ▪ Warning - Warning message ▪ Error - Indicates the process has failed ▪ Fatal ▪ Control - An expected change in state such as the start or completion of an archival, log recovery, or switchover operation
DEST_ID	NUMBER	Destination ID number to which the event pertains. If the event does not pertain to a particular destination, then the value is 0.
MESSAGE_NUM	NUMBER	A chronologically increasing number giving each event a unique number
ERROR_CODE	NUMBER	Error ID pertaining to the event
CALLOUT	VARCHAR2(3)	Indicates whether the current entry is a callout event (YES) or not (NO) A YES value in this column indicates that this event may require the DBA to perform some action. Examine the ERROR_CODE and MESSAGE columns for more information. A NO value in this column generally corresponds to an INFORMATIONAL or WARNING event which would not require any action by the DBA.
TIMESTAMP	DATE	Time stamp

Column	Datatype	Description
MESSAGE	VARCHAR2(256)	A text message describing the event

V\$DB_CACHE_ADVICE

V\$DB_CACHE_ADVICE contains rows that predict the number of physical reads for the cache size corresponding to each row. The rows also compute a "physical read factor," which is the ratio of the number of estimated reads to the number of reads actually performed by the real buffer cache during the measurement interval .

See Also: ["DB_CACHE_ADVICE"](#) on page 1-35

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier (ranges from 1 to 8)
NAME	VARCHAR2(20)	Buffer pool name
BLOCK_SIZE	NUMBER	Block size in bytes for buffers in this pool. Possible values: the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, 32768.
ADVICE_STATUS	VARCHAR2(3)	Status of the advisory. ON indicates it is currently running; OFF indicates it is disabled (in this case the estimates are historical and calculated when last enabled).
SIZE_FOR_ESTIMATE	NUMBER	Cache size for prediction (in megabytes)
SIZE_FACTOR	NUMBER	Size factor with respect to the current cache size
BUFFERS_FOR_ESTIMATE	NUMBER	Cache size for prediction (in terms of buffers)
ESTD_PHYSICAL_READ_FACTOR	NUMBER	Physical read factor for this cache size, which is the ratio of the number of estimated physical reads to the number of reads in the real cache. If there are no physical reads in the real cache, the value of this column is null.
ESTD_PHYSICAL_READS	NUMBER	Estimated number of physical reads for this cache size
ESTD_PHYSICAL_READ_TIME	NUMBER	Estimated disk read time
ESTD_PCT_OF_DB_TIME_FOR_READS	NUMBER	Estimated disk time as a percentage of the total time

V\$DB_OBJECT_CACHE

This view displays database objects that are cached in the library cache. Objects include tables, indexes, clusters, synonym definitions, PL/SQL procedures and packages, and triggers.

Column	Datatype	Description
OWNER	VARCHAR2(64)	Owner of the object
NAME	VARCHAR2(1000)	Name of the object
DB_LINK	VARCHAR2(64)	Database link name, if any
NAMESPACE	VARCHAR2(28)	Library cache namespace of the object: TABLE/PROCEDURE, BODY, TRIGGER, INDEX, CLUSTER, OBJECT
TYPE	VARCHAR2(28)	Type of the object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, DBLINK
SHARABLE_MEM	NUMBER	Amount of sharable memory in the shared pool consumed by the object
LOADS	NUMBER	Number of times the object has been loaded. This count also increases when an object has been invalidated.

Column	Datatype	Description
EXECUTIONS	NUMBER	Not used See Also: " V\$SQLAREA " on page 5-68 to see actual execution counts
LOCKS	NUMBER	Number of users currently locking this object
PINS	NUMBER	Number of users currently pinning this object
KEPT	VARCHAR2(3)	(YES NO) Depends on whether this object has been "kept" (permanently pinned in memory) with the PL/SQL procedure <code>DBMS_SHARED_POOL.KEEP</code>
CHILD_LATCH	NUMBER	Child latch number that is protecting the object

V\$DB_PIPES

This view displays the pipes that are currently represented in the shared pool for this instance.

Column	Datatype	Description
OWNERID	NUMBER	The owner ID of the owner if this is a private pipe. NULL otherwise.
NAME	VARCHAR2(1000)	The name of the pipe (for example, SCOTT.PIPE)
TYPE	VARCHAR2(7)	PUBLIC or PRIVATE
PIPE_SIZE	NUMBER	The amount of memory the pipe uses

V\$DBFILE

This view lists all datafiles making up the database. This view is retained for historical compatibility. Use of `V$DATAFILE` is recommended instead.

Column	Datatype	Description
FILE#	NUMBER	File identifier
NAME	VARCHAR2(513)	Name of the file

See Also: "[V\\$DATAFILE](#)" on page 4-44

V\$DBLINK

This view describes all database links (links with `IN_TRANSACTION = YES`) opened by the session issuing the query on `V$DBLINK`. These database links must be committed or rolled back before being closed.

Column	Datatype	Description
DB_LINK	VARCHAR2(128)	Name of the database link
OWNER_ID	NUMBER	Owner of the database link UID
LOGGED_ON	VARCHAR2(3)	Whether the database link is currently logged on
HETEROGENEOUS	VARCHAR2(3)	Whether the database link is heterogeneous
PROTOCOL	VARCHAR2(6)	Communication protocol for the database link
OPEN_CURSORS	NUMBER	Whether there are open cursors for the database link
IN_TRANSACTION	VARCHAR2(3)	Whether the database link is currently in a transaction
UPDATE_SENT	VARCHAR2(3)	Whether there has been an update on the database link
COMMIT_POINT_STRENGTH	NUMBER	Commit point strength of the transactions on the database link

V\$DELETED_OBJECT

This view displays information about deleted archived logs, datafile copies and backup pieces from the control file. The only purpose of this view is to optimize the recovery catalog resync operation. When an archived log, datafile copy, or backup piece is deleted, the corresponding record is marked deleted.

Column	Datatype	Description
RECID	NUMBER	Deleted object record ID
STAMP	NUMBER	Deleted object record stamp
TYPE	VARCHAR2(26)	Identifies the type of deleted object: <ul style="list-style-type: none"> ▪ ARCHIVED LOG ▪ BACKUP PIECE ▪ DATAFILE COPY ▪ PROXY COPY ▪ BACKUP PIECE AVAILABLE ▪ BACKUP PIECE EXPIRED ▪ PROXY COPY AVAILABLE ▪ PROXY COPY EXPIRED ▪ BACKUP PIECE UNAVAILABLE ▪ PROXY COPY UNAVAILABLE ▪ DATAFILE COPY AVAILABLE ▪ DATAFILE COPY EXPIRED ▪ DATAFILE COPY UNAVAILABLE ▪ ARCHIVED LOG AVAILABLE ▪ ARCHIVED LOG EXPIRED ▪ ARCHIVED LOG UNAVAILABLE ▪ BACKUP SET KEEP OPTIONS ▪ BACKUP SET KEEP UNTIL ▪ PROXY COPY KEEP OPTIONS ▪ PROXY COPY KEEP UNTIL ▪ DATAFILE COPY KEEP OPTIONS ▪ DATAFILE COPY KEEP UNTIL ▪ DATAFILE RENAME ON RESTORE
OBJECT_RECID	NUMBER	Record ID of the deleted object
OBJECT_STAMP	NUMBER	Record timestamp of the deleted object
OBJECT_DATA	NUMBER	Displays additional internal information related to this deleted object. For internal Oracle use only.

V\$DISPATCHER

V\$DISPATCHER displays information about the dispatcher processes.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the dispatcher process
NETWORK	VARCHAR2(128)	Network address of the dispatcher
PADDR	RAW(4 8)	Process address

Column	Datatype	Description
STATUS	VARCHAR2 (16)	Status of the dispatcher: <ul style="list-style-type: none"> ▪ WAIT - Idle ▪ SEND - Sending a message ▪ RECEIVE - Receiving a message ▪ CONNECT - Establishing a connection ▪ DISCONNECT - Handling a disconnect request ▪ BREAK - Handling a break ▪ TERMINATE - In the process of terminating ▪ ACCEPT - Accepting connections (no further information available) ▪ REFUSE - Rejecting connections (no further information available)
ACCEPT	VARCHAR2 (3)	Indicates whether the dispatcher is accepting new connections (YES) or not (NO)
MESSAGES	NUMBER	Number of messages processed by the dispatcher
BYTES	NUMBER	Size (in bytes) of messages processed by the dispatcher
BREAKS	NUMBER	Number of breaks occurring in the connection
OWNED	NUMBER	Number of circuits owned by the dispatcher
CREATED	NUMBER	Number of circuits created by the dispatcher
IDLE	NUMBER	Total idle time for the dispatcher (in hundredths of a second)
BUSY	NUMBER	Total busy time for the dispatcher (in hundredths of a second)
LISTENER	NUMBER	Most recent Oracle error number the dispatcher received from the listener
CONF_INDX	NUMBER	Zero-based index of the DISPATCHERS configuration used by the dispatcher

V\$DISPATCHER_CONFIG

V\$DISPATCHER_CONFIG displays information about the dispatcher configurations and their attributes.

Column	Datatype	Description
CONF_INDX	NUMBER	Zero-based index of the DISPATCHERS configuration
NETWORK	VARCHAR2 (128)	Network protocol or listening address of dispatchers (may be truncated)
DISPATCHERS	NUMBER	Number of dispatchers to maintain for the configuration
CONNECTIONS	NUMBER	Maximum number of concurrent connections per dispatcher
SESSIONS	NUMBER	Maximum number of concurrent sessions per dispatcher
POOL	VARCHAR2 (4)	Indicates whether Connection Pooling is on: <ul style="list-style-type: none"> ▪ IN ▪ OUT ▪ BOTH ▪ OFF
TICKS	NUMBER	Tick size (in seconds) for Connection Pooling timeout
INBD_TIMEOUT	NUMBER	Timeout duration (in ticks) for pooled inbound connections
OUTBD_TIMEOUT	NUMBER	Timeout duration (in ticks) for pooled outbound connections
MULTIPLY	VARCHAR2 (4)	Indicates whether Session Multiplexing is on: <ul style="list-style-type: none"> ▪ IN ▪ OUT ▪ BOTH ▪ OFF

Column	Datatype	Description
LISTENER	VARCHAR2(1200)	Listeners to register dispatchers with (may be truncated)
SERVICE	VARCHAR2(128)	Service names supported (may be truncated)

V\$DISPATCHER_RATE

V\$DISPATCHER_RATE displays rate statistics for a number of activities performed by the dispatcher processes. Collected samples have an activity-specific "time-to-live" (TTL_* columns). Statistics are reported over the following two types of time intervals:

- Current statistics (CUR_ columns)

Current statistics use samples collected over the most recent time-to-live interval.
- Historical statistics (AVG_ and most of the MAX_ columns)

Historical statistics make use of all samples that are no longer current.

At the time of collection, a sample is current. After the time-to-live has elapsed, the sample becomes historical. Each type of activity has a specific scale (represented by the SCALE_* columns) at which the statistics are reported.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the dispatcher process
PADDR	RAW(4 8)	Address of the dispatcher process
CUR_LOOP_RATE	NUMBER	Rate at which the dispatcher has been iterating through its dispatching loop, reported over the past TTL_LOOPS, in iterations/SCALE_LOOPS
CUR_EVENT_RATE	NUMBER	Rate at which the dispatcher has been processing dispatcher events, reported over the past TTL_LOOPS, in events/SCALE_LOOPS. Such dispatcher events include network events and shared server requests.
CUR_EVENTS_PER_LOOP	NUMBER	Average number of events the dispatcher has been processing in each iteration through its dispatching loop, reported over the past TTL_LOOPS, in events/iteration
CUR_MSG_RATE	NUMBER	Rate at which the dispatcher has been relaying messages between clients and shared servers, reported over the past TTL_MSG, in messages/SCALE_MSG
CUR_SVR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to shared servers, reported over the past TTL_SVR_BUF, in messages/SCALE_SVR_BUF
CUR_SVR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to shared servers, reported over the past TTL_SVR_BUF, in bytes/SCALE_SVR_BUF
CUR_SVR_BYTE_PER_BUF	NUMBER	Average number of data types in each buffer relayed to shared servers, reported over the past TTL_SVR_BUF, in bytes/buffer
CUR_CLT_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to clients, reported over the past TTL_CLT_BUF, in buffers/SCALE_CLT_BUF
CUR_CLT_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to clients, reported over the past TTL_CLT_BUF, in bytes/SCALE_CLT_BUF
CUR_CLT_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to clients, reported over the past TTL_CLT_BUF, in bytes/buffer
CUR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to either clients or shared servers, reported over the past TTL_BUF, in bytes/SCALE_BUF
CUR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to either clients or shared servers, reported over the past TTL_BUF, in bytes/SCALE_BUF
CUR_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to either clients or shared servers, reported over the past TTL_BUF, in bytes/buffer

Column	Datatype	Description
CUR_IN_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been accepting incoming client connections, reported over the past TTL_IN_CONNECT, in connections/SCALE_IN_CONNECT
CUR_OUT_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been establishing outbound connections, reported over the past TTL_OUT_CONNECT, in connections/SCALE_OUT_CONNECT
CUR_RECONNECT_RATE	NUMBER	In a connection pooling setup, the rate at which clients have been reconnecting to the dispatcher, reported over the past TTL_RECONNECT, in reconconnections/SCALE_RECONNECT
MAX_LOOP_RATE	NUMBER	Maximum rate at which the dispatcher has ever iterated through its dispatching loop, reported in iterations/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENT_RATE	NUMBER	Maximum rate at which the dispatcher has ever processed dispatcher events, reported in events/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENTS_PER_LOOP	NUMBER	Maximum number of events the dispatcher has ever processed in one iteration through its dispatching loop, reported in events/iteration, over the dispatcher's lifetime
MAX_MSG_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed messages between clients and shared servers, reported in messages/SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
MAX_SVR_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to shared servers, reported in buffers/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to shared servers, reported in bytes/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes/buffer, over the dispatcher's lifetime
MAX_CLT_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers/SCALE_CLT_BUF, over the dispatcher's life time excluding the past TTL_CLT_BUF
MAX_CLT_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to clients, reported in bytes/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the last TTL_CLT_BUF
MAX_CLT_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes/buffer, over the dispatcher's lifetime
MAX_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers/SCALE_BUF, over the dispatcher's lifetime, excluding the past TTL_BUF
MAX_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to either clients or shared servers, reported in bytes/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
MAX_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to either a client or a shared server, reported in bytes/buffer, over the dispatcher's lifetime
MAX_IN_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever accepted incoming client connections, reported in connections/SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT
MAX_OUT_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever established outbound connections, reported in connections/SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT
MAX_RECONNECT_RATE	NUMBER	In a connection pooling setup, the maximum rate at which clients have ever reconnected to this dispatcher, reported in reconconnections/SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
AVG_LOOP_RATE	NUMBER	Historical average rate at which the dispatcher has iterated through its dispatching loop, reported in iterations/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS

Column	Datatype	Description
AVG_EVENT_RATE	NUMBER	Historical average rate at which the dispatcher has processed dispatcher events, reported in events/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_EVENTS_PER_LOOP	NUMBER	Historical average number of events the dispatcher has processed in one iteration through its dispatching loop, reported in events/iteration, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_MSG_RATE	NUMBER	Historical average rate at which the dispatcher has relayed messages between clients and shared servers, reported in messages/SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
AVG_SVR_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to shared servers, reported in buffers/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to shared servers, reported in bytes/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to shared servers, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_CLT_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to clients, reported in buffers/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to clients, reported in bytes/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to clients, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to either clients or shared servers, reported in buffers/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to either clients or shared servers, reported in bytes/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to either clients or shared servers, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_IN_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has accepted incoming client connections, reported in connections/SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT
AVG_OUT_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has established outbound connections, reported in connections/SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT
AVG_RECONNECT_RATE	NUMBER	In a connection pooling setup, the historical average rate at which clients have reconnected to this dispatcher, reported in reconnections/SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
TTL_LOOPS	NUMBER	Time-to-live for "loops" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_MSG	NUMBER	Time-to-live for "messages" samples, reported in hundredths of a second. Default is 10 seconds.
TTL_SVR_BUF	NUMBER	Time-to-live for "buffers to servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_CLT_BUF	NUMBER	Time-to-live for "buffers to clients" samples, reported in hundredths of a second. Default is 1 second.
TTL_BUF	NUMBER	Time-to-live for "buffers to clients/servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_IN_CONNECT	NUMBER	Time-to-live for "inbound connections" samples, reported in hundredths of a second. Default is 10 minutes.

Column	Datatype	Description
TTL_OUT_CONNECT	NUMBER	Time-to-live for "outbound connections" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_RECONNECT	NUMBER	Time-to-live for "reconnections" samples, reported in hundredths of a second. Default is 10 minutes.
SCALE_LOOPS	NUMBER	Scale for "loops" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_MSG	NUMBER	Scale for "messages" statistics, reported in hundredths of a second. Default is 1 second.
SCALE_SVR_BUF	NUMBER	Scale for "buffers to servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_CLT_BUF	NUMBER	Scale for "buffers to clients" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_BUF	NUMBER	Scale for "buffers to clients/servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_IN_CONNECT	NUMBER	Scale for "inbound connections" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_OUT_CONNECT	NUMBER	Scale for "outbound connections" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_RECONNECT	NUMBER	Scale for "reconnections" statistics, reported in hundredths of a second. Default is 1 minute.

V\$ENABLEDPRIVS

This view displays which privileges are enabled. These privileges can be found in the table SYS.SYSTEM_PRIVILEGES_MAP.

Column	Datatype	Description
PRIV_NUMBER	NUMBER	Numeric identifier of enabled privileges

V\$ENQUEUE_LOCK

This view displays all locks owned by enqueue state objects. The columns in this view are identical to the columns in V\$LOCK.

See Also: "[V\\$LOCK](#)" on page 4-76

Column	Datatype	Description
ADDR	RAW(4 8)	Address of lock state object
KADDR	RAW(4 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. Lists user and system types that can have locks.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)

Column	Datatype	Description
LMODE	NUMBER	Lock mode in which the session holds the lock: <ul style="list-style-type: none"> ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: <ul style="list-style-type: none"> ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V\$ENQUEUE_STAT

This view displays statistics on the number of enqueue (lock) requests for each type of lock.

Column	Datatype	Description
INST_ID	NUMBER	Database instance number
EQ_TYPE	VARCHAR2(2)	Type of enqueue requested
TOTAL_REQ#	NUMBER	Total number of enqueue requests or enqueue conversions for this type of enqueue
TOTAL_WAIT#	NUMBER	Total number of times an enqueue request or conversion resulted in a wait
SUCC_REQ#	NUMBER	Number of times an enqueue request or conversion was granted
FAILED_REQ#	NUMBER	Number of times an enqueue request or conversion failed
CUM_WAIT_TIME	NUMBER	Total amount of time (in milliseconds) spent waiting for the enqueue or enqueue conversion

V\$EVENT_HISTOGRAM

V\$EVENT_HISTOGRAM displays a histogram of the number of waits, the maximum wait, and total wait time on an event basis. The histogram has buckets of time intervals from < 1 ms, < 2 ms, < 4 ms, < 8 ms, ... < 2²¹ ms, < 2²² ms, >= 2²² ms.

The histogram will not be filled unless the TIMED_STATISTICS initialization parameter is set to true.

Column	Datatype	Description
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Name of the Event
WAIT_TIME_MILLI	NUMBER	Amount of time the bucket represents (in milliseconds). If the duration = num, then this column represents waits of duration < num that are not included in any smaller bucket.

Column	Datatype	Description
WAIT_COUNT	NUMBER	Number of waits of the duration belonging to the bucket of the histogram

V\$EVENT_NAME

V\$EVENT_NAME displays information about wait events.

Column	Datatype	Description
EVENT#	NUMBER	Number of the wait event
EVENT_ID	NUMBER	Identifier of the wait event
NAME	VARCHAR2 (64)	Name of the wait event
PARAMETER1	VARCHAR2 (64)	Description of the first parameter for the wait event
PARAMETER2	VARCHAR2 (64)	Description of the second parameter for the wait event
PARAMETER3	VARCHAR2 (64)	Description of the third parameter for the wait event
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2 (64)	Name of the class of the wait event

V\$EVENTMETRIC

V\$EVENTMETRIC displays values of wait event metrics for the most recent 60-second interval.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
EVENT#	NUMBER	Number of the event
EVENT_ID	NUMBER	Identifier of the event
NUM_SESS_WAITING	NUMBER	Number of sessions waiting at the end of the interval
TIME_WAITED	NUMBER	Time waited
WAIT_COUNT	NUMBER	Number of times waited

V\$EXECUTION

This view displays information on parallel execution.

Column	Datatype	Description
PID	NUMBER	Session ID
DEPTH	NUMBER	The depth
FUNCTION	VARCHAR2 (10)	Session serial number
TYPE	VARCHAR2 (7)	Name of the OBJECT_NODE in plan table
NVALS	NUMBER	Elapsed time for OBJECT_NODE
VAL1	NUMBER	The value for number 1
VAL2	NUMBER	The value for number 2
SEQH	NUMBER	A sequence
SEQL	NUMBER	A sequence

V\$FALSE_PING

V\$FALSE_PING is a Real Application Clusters view. This view displays buffers that may be getting false pings. That is, buffers pinged more than 10 times that are protected by the same lock as another buffer that pinged more than 10 times. Buffers identified as getting false pings can be remapped in "[GC_FILES_TO_LOCKS](#)" to reduce lock collisions.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V\$DBFILE)
BLOCK#	NUMBER	Block number
STATUS	VARCHAR2(6)	Status of the block: <ul style="list-style-type: none"> ▪ free - Not currently in use ▪ xcur - Exclusive ▪ scur - Shared current ▪ cr - Consistent read ▪ read - Being read from disk ▪ mrec - In media recovery mode ▪ irec - In instance recovery mode
XNC	NUMBER	Number of PCM lock conversions from Exclusive mode due to contention with another instance. This column is obsolete and maintained for backward compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	NULL for non-partitioned objects
KIND	VARCHAR2(15)	Type of database object
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Name of the lock that contains the PCM lock that is covering the buffer
LOCK_ELEMENT_CLASS	NUMBER	Lock element class

See Also: "[GC_FILES_TO_LOCKS](#)" on page 1-56 and *Oracle Real Application Clusters Installation and Configuration Guide*

V\$FAST_START_SERVERS

V\$FAST_START_SERVERS provides information about all the recovery slaves performing parallel transaction recovery.

Column	Datatype	Description
STATE	VARCHAR2(11)	State of the server (IDLE or RECOVERING)
UNDOBLOCKSDONE	NUMBER	Number of undo blocks done so far
PID	NUMBER	Process ID
XID	RAW(8)	Transaction ID

See Also: *Oracle Database Performance Tuning Guide*

V\$FAST_START_TRANSACTIONS

V\$FAST_START_TRANSACTIONS displays information about the progress of the transactions that Oracle is recovering.

Column	Datatype	Description
USN	NUMBER	Undo segment number of the transaction
SLT	NUMBER	Slot within the rollback segment
SEQ	NUMBER	Incarnation number of the slot
STATE	VARCHAR2(16)	State of the transaction (may be TO BE RECOVERED, RECOVERED, or RECOVERING)
UNDOBLOCKSDONE	NUMBER	Number of undo blocks completed on the transaction
UNDOBLOCKSTOTAL	NUMBER	Total number of undo blocks that need recovery
PID	NUMBER	ID of the current server it has been assigned to
CPUTIME	NUMBER	Time for which recovery has progressed (in seconds)
PARENTUSN	NUMBER	Undo segment number of the parent transaction in PDML
PARENTSLT	NUMBER	Slot of the parent transaction in PDML
PARENTSEQ	NUMBER	Sequence number of the parent transaction in PDML
XID	RAW(8)	Transaction ID
PXID	RAW(8)	Parent transaction ID
RCVSERVERS	NUMBER	Number of servers used in the last recovery

See Also: *Oracle Database Performance Tuning Guide*

V\$FILE_CACHE_TRANSFER

V\$FILE_CACHE_TRANSFER is deprecated.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
RBR	NUMBER	Number of reuse blocks cross-instance calls; always 0
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse blocks cross-instance calls; always 0
RBR_FORCED_STALE	NUMBER	Number of blocks marked as flushed due to reuse blocks cross-instance calls; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0

Column	Datatype	Description
CR_TRANSFERS	NUMBER	Number of CR blocks received; always 0
CUR_TRANSFERS	NUMBER	Number of current blocks received; always 0

V\$FILE_HISTOGRAM

V\$FILE_HISTOGRAM displays a histogram of all single block reads on a per-file basis. The histogram has buckets of time intervals from < 1 ms, < 2 ms, < 4 ms, < 8 ms, ... < 2²¹ ms, < 2²² ms, >= 2²² ms.

The histogram will not be filled unless the TIMED_STATISTICS initialization parameter is set to true.

Column	Datatype	Description
FILE#	NUMBER	File number
SINGLEBLKRDTIM_MILLI	NUMBER	Amount of time the bucket represents (in milliseconds). If the duration = <i>num</i> , then this column represents waits of duration < <i>num</i> that are not included in any smaller bucket.
SINGLEBLKRDS	NUMBER	Number of waits of the duration belonging to the bucket of the histogram

V\$FILEMETRIC

V\$FILEMETRIC displays values of file metrics for the most recent 10-minute interval. A history of the last one hour will be kept in the system.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
FILE_ID	NUMBER	File number
CREATION_TIME	NUMBER	Timestamp of the file creation
AVERAGE_READ_TIME	NUMBER	Average file read time
AVERAGE_WRITE_TIME	NUMBER	Average file write time
PHYSICAL_READS	NUMBER	Number of physical reads
PHYSICAL_WRITES	NUMBER	Number of physical writes
PHYSICAL_BLOCK_READS	NUMBER	Number of physical block reads
PHYSICAL_BLOCK_WRITES	NUMBER	Number of physical block writes

V\$FILEMETRIC_HISTORY

V\$FILEMETRIC_HISTORY displays values of file metrics for all intervals in the last one hour.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
FILE_ID	NUMBER	File number
CREATION_TIME	NUMBER	Timestamp of the file creation

Column	Datatype	Description
AVERAGE_READ_TIME	NUMBER	Average file read time
AVERAGE_WRITE_TIME	NUMBER	Average file write time
PHYSICAL_READS	NUMBER	Number of physical reads
PHYSICAL_WRITES	NUMBER	Number of physical writes
PHYSICAL_BLOCK_READS	NUMBER	Number of physical block reads
PHYSICAL_BLOCK_WRITES	NUMBER	Number of physical block writes

V\$FILESTAT

This view contains information about file read/write statistics.

Column	Datatype	Description
FILE#	NUMBER	Number of the file
PHYRDS	NUMBER	Number of physical reads done
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
SINGLEBLKRDS	NUMBER	Number of single block reads
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER	Cumulative single block read time (in hundredths of a second)
AVGIOTIM	NUMBER	Average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is true; 0 if false
LSTIOTIM	NUMBER	Time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MINIOTIM	NUMBER	Minimum time (in hundredths of a second) spent on a single I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIORTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is true; 0 if false

V\$FIXED_TABLE

This view displays all dynamic performance tables, views, and derived tables in the database. Some V\$ tables (for example, V\$ROLLNAME) refer to real tables and are therefore not listed.

Column	Datatype	Description
NAME	VARCHAR2 (30)	Name of the object
OBJECT_ID	NUMBER	Identifier of the fixed object
TYPE	VARCHAR2 (5)	Object type (TABLE VIEW)
TABLE_NUM	NUMBER	Number that identifies the dynamic performance table if it is of type TABLE

V\$FIXED_VIEW_DEFINITION

This view contains the definitions of all the fixed views (views beginning with V\$). Use this table with caution. Oracle tries to keep the behavior of fixed views the same from release to release, but the definitions of the fixed views can change without notice. Use these definitions to optimize your queries by using indexed columns of the dynamic performance tables.

Column	Datatype	Description
VIEW_NAME	VARCHAR2(30)	Name of the fixed view
VIEW_DEFINITION	VARCHAR2(4000)	Definition of the fixed view

V\$FLASHBACK_DATABASE_LOG

V\$FLASHBACK_DATABASE_LOG displays information about the flashback data. Use this view to help estimate the amount of flashback space required for the current workload.

Column	Datatype	Description
OLDEST_FLASHBACK_SCN	NUMBER	Lowest system change number (SCN) in the flashback data
OLDEST_FLASHBACK_TIME	DATE	Time of the lowest SCN in the flashback data
RETENTION_TARGET	NUMBER	Target retention time (in minutes)
FLASHBACK_SIZE	NUMBER	Current size (in bytes) of the flashback data
ESTIMATED_FLASHBACK_SIZE	NUMBER	Estimated size of flashback data needed for the current target retention

V\$FLASHBACK_DATABASE_STAT

V\$FLASHBACK_DATABASE_STAT displays statistics for monitoring the I/O overhead of logging flashback data. This view also displays the estimated flashback space needed based on previous workloads.

Column	Datatype	Description
BEGIN_TIME	DATE	Beginning of the time interval
END_TIME	DATE	End of the time interval
FLASHBACK_DATA	NUMBER	Number of bytes of flashback data written during the interval
DB_DATA	NUMBER	Number of bytes of database data read and written during the interval
REDO_DATA	NUMBER	Number of bytes of redo data written during the interval
ESTIMATED_FLASHBACK_SIZE	NUMBER	Value of ESTIMATED_FLASHBACK_SIZE in V\$FLASHBACK_DATABASE_LOG at the end of the time interval

V\$GC_ELEMENT

This is a Real Application Clusters view. There is one entry in V\$GC_ELEMENT for each global cache resource that is used by the buffer cache. The name of the global cache resource that corresponds to a lock element is {'BL', indx, class}.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
INDX	NUMBER	Platform specific lock manager identifier
CLASS	NUMBER	Platform specific lock manager identifier
GC_ELEMENT_NAME	NUMBER	Name of the lock that contains the PCM lock that is covering the buffer
MODE_HELD	NUMBER	Platform dependent value for lock mode held; often: 3 = share; 5 = exclusive
BLOCK_COUNT	NUMBER	Number of blocks covered by PCM lock
RELEASING	NUMBER	Nonzero if PCM lock is being downgraded
ACQUIRING	NUMBER	Nonzero if PCM lock is being upgraded
INVALID	NUMBER	Nonzero if PCM lock is invalid (a lock may become invalid after a system failure)
FLAGS	NUMBER	Process level flags for the lock element

V\$GC_ELEMENTS_WITH_COLLISIONS

This is a Real Application Clusters view. Use this view to find the locks that protect multiple buffers, each of which has been either force-written or force-read at least 10 times. It is very likely that those buffers are experiencing false pings due to being mapped to the same lock.

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4 8)	Address of the lock element that contains the PCM lock covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.

V\$GCSHVMMASTER_INFO

V\$GCSHVMMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Cache Service resources except those belonging to files mapped to a particular master.

Column	Datatype	Description
HV_ID	NUMBER	PCM hash value ID
CURRENT_MASTER	NUMBER	Master instance of this PCM hash value ID
PREVIOUS_MASTER	NUMBER	Previous master instance of this PCM hash value ID
REMASTER_CNT	NUMBER	Number of times this has been remastered

V\$GCSPFMASTER_INFO

V\$GCSPFMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Cache Service resources belonging to files mapped to instances.

Column	Datatype	Description
FILE_ID	NUMBER	File number
CURRENT_MASTER	NUMBER	Master instance of this file
PREVIOUS_MASTER	NUMBER	Previous master instance of this file
REMASTER_CNT	NUMBER	Number of times this has been remastered

V\$GES_BLOCKING_ENQUEUE

This Real Application Clusters view describes all locks currently known to lock manager that are being blocked or blocking others. The output of this view is a subset of the output from V\$GES_ENQUEUE.

See Also: "V\$GES_ENQUEUE" on page 4-65 for a description of all locks known to the lock manager

Column	Datatype	Description
HANDLE	RAW(4 8)	Lock pointer
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier where the lock belongs to
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier where the lock belongs to
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS_OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT_GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT_PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT_NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT_DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT_NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue.
STATE	VARCHAR2(64)	State of lock as owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V\$GES_CONVERT_LOCAL

V\$GES_CONVERT_LOCAL displays average convert time, count information, and timed statistics for remote GES enqueue operations.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
CONVERT_TYPE	VARCHAR2(64)	Conversion types are listed in Table 5-5
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in hundredths of a second)
CONVERT_COUNT	NUMBER	The number of operations

V\$GES_CONVERT_REMOTE

V\$GES_CONVERT_REMOTE displays average convert time, count information, and timed statistics for remote GES enqueue operations.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
CONVERT_TYPE	VARCHAR2(64)	Conversion types are listed in Table 5-5 <ul style="list-style-type: none"> ▪ NULL -> SS - NULL mode to subshared mode ▪ NULL -> SX - NULL mode to shared exclusive mode ▪ NULL -> S - NULL mode to shared mode ▪ NULL -> SSX - NULL mode to subshared exclusive mode ▪ NULL -> X - NULL mode to exclusive mode ▪ SS -> SX - Subshared mode to shared exclusive mode ▪ SS -> S - Subshared mode to shared mode ▪ SS -> SSX - Subshared mode to subshared exclusive mode ▪ SS -> X - Subshared mode to exclusive mode ▪ SX -> S - Shared exclusive mode to shared mode ▪ SX -> SSX - Shared exclusive mode to subshared exclusive mode ▪ SX -> X - Shared exclusive mode to exclusive mode ▪ S -> SX - Shared mode to shared exclusive mode ▪ S -> SSX - Shared mode to subshared exclusive mode ▪ S -> X - Shared mode to exclusive mode ▪ SSX -> X - Sub-shared exclusive mode to exclusive mode
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in hundredths of a second)
CONVERT_COUNT	NUMBER	Number of operations

V\$GES_ENQUEUE

This Real Application Clusters view describes all locks currently known to lock manager.

See Also: "[V\\$GES_BLOCKING_ENQUEUE](#)" on page 4-64 for a description of all such locks that are currently blocking or being blocked

Column	Datatype	Description
HANDLE	RAW(4 8)	Lock pointer

V\$GES_LATCH

Column	Datatype	Description
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier to which the lock belongs
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier to which the lock belongs
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS_OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT_GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT_PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT_NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT_DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT_NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue.
STATE	VARCHAR2(64)	State of the lock as the owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V\$GES_LATCH

V\$GES_LATCH is obsolete.

See Also: ["V\\$LATCH"](#) on page 4-72 for statistics about GES latch performance

V\$GES_RESOURCE

V\$GES_RESOURCE is a Real Application Clusters view. It displays information of all resources currently known to the lock manager.

Column	Datatype	Description
RESP	RAW(4 8)	Resource pointer
RESOURCE_NAME	VARCHAR2(30)	Resource name in hexadecimal for the lock
ON_CONVERT_Q	NUMBER	1 if on convert queue, 0 otherwise

Column	Datatype	Description
ON_GRANT_Q	NUMBER	1 if on granted queue, 0 otherwise
PERSISTENT_RES	NUMBER	1 if it is a persistent resource, 0 otherwise
RDOMAIN_NAME	VARCHAR2(25)	Recovery domain name
RDOMAINP	RAW(4 8)	Recovery domain pointer
MASTER_NODE	NUMBER	Master node ID
NEXT_CVT_LEVEL	VARCHAR2(9)	Next lock level to convert on global convert queue
VALUE_BLK_STATE	VARCHAR2(32)	State of the value block
VALUE_BLK	VARCHAR2(64)	First 64 bytes of the value block

V\$GES_STATISTICS

V\$GES_STATISTICS displays miscellaneous GES statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Name of the statistic
VALUE	NUMBER	Value associated with the statistic

V\$GLOBAL_BLOCKED_LOCKS

This view displays global blocked locks.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of lock state object
KADDR	RAW(4 8)	Address of lock
SID	NUMBER	Identifier of session holding the lock (number)
TYPE	VARCHAR2(2)	Resource type (char)
ID1	NUMBER	Resource identifier #1 (number)
ID2	NUMBER	Resource identifier #2 (number)
LMODE	NUMBER	Lock mode held (number)
REQUEST	NUMBER	Lock mode requested (number)
CTIME	NUMBER	Time since current mode was granted

V\$GLOBAL_TRANSACTION

This view displays information on the currently active global transactions.

Column	Datatype	Description
FORMATID	NUMBER	Format identifier of the global transaction
GLOBALID	RAW(64)	Global transaction identifier of the global transaction
BRANCHID	RAW(64)	Branch qualifier of the global transaction
BRANCHES	NUMBER	Total number of branches in the global transaction
REFCOUNT	NUMBER	Number of siblings for this global transaction, must be the same as branches
PREPARECOUNT	NUMBER	Number of branches of the global transaction that have prepared
STATE	VARCHAR2(18)	State of the branch of the global transaction

Column	Datatype	Description
FLAGS	NUMBER	The numerical representation of the state
COUPLING	VARCHAR2(15)	Whether the branches are loosely coupled or tightly coupled

V\$HS_AGENT

This view identifies the set of HS agents currently running on a given host, using one row per agent process.

Column	Datatype	Description
AGENT_ID	NUMBER	Oracle Net session identifier used for connections to agent (the identifier used in the LISTENER.ORA file). Maps to the AGENT_ID column of V\$HS_AGENT.
MACHINE	VARCHAR2(64)	Operating system machine name
PROCESS	VARCHAR2(9)	Operating system process identifier of agent
PROGRAM	VARCHAR2(48)	Program name of agent
OSUSER	VARCHAR2(30)	Operating system user
STARTTIME	DATE	The starting time
AGENT_TYPE	NUMBER	Type of agent
FDS_CLASS_ID	NUMBER	The ID of the Foreign Data Store class
FDS_INST_ID	NUMBER	The instance name of the Foreign Data Store

V\$HS_PARAMETER

This view describes the initialization parameters in use by the server and agent.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier. This column maps to the HS_SESSION_ID column of V\$HS_SESSION.
PARAMETER	VARCHAR2(30)	Name of the parameter
VALUE	VARCHAR2(64)	Value of the parameter
SOURCE	VARCHAR2(1)	Whether the parameter was defined in the agent (A) or server (S)
ENV	VARCHAR2(1)	Whether the parameter was also set as an environment variable in the agent or elsewhere (T F)

V\$HS_SESSION

This view describes the current HS session.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier
AGENT_ID	NUMBER	Oracle Net session identifier used for connections to the agent. Maps to the AGENT_ID column of V\$HS_AGENT.
SID	NUMBER	User session identifier. Maps to the SID column of V\$SESSION.
DB_LINK	VARCHAR2(128)	Server database link name used to access the agent. Blank if no database link is used (for example, when using external procedures).
DB_LINK_OWNER	NUMBER	Owner of the database link in DB_LINK
STARTTIME	DATE	Time the connection was initiated

V\$HVMMASTER_INFO

V\$HVMMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Enqueue Service resources.

Column	Datatype	Description
HV_ID	NUMBER	Hash value ID
CURRENT_MASTER	NUMBER	Master instance of this hash value ID
PREVIOUS_MASTER	NUMBER	Previous master instance of this hash value ID
REMASTER_CNT	NUMBER	Number of times this has been remastered

V\$INDEXED_FIXED_COLUMN

This view displays the columns in dynamic performance tables that are indexed (X\$ tables). The X\$ tables can change without notice. Use this view only to write queries against fixed views (V\$ views) more efficiently.

Column	Datatype	Description
TABLE_NAME	VARCHAR2(30)	Name of the dynamic performance table that is indexed
INDEX_NUMBER	NUMBER	Number that distinguishes to which index a column belongs
COLUMN_NAME	VARCHAR2(30)	Name of the column that is being indexed
COLUMN_POSITION	NUMBER	Position of the column in the index key (this is mostly relevant for multicolumn indexes)

V\$INSTANCE

This view displays the state of the current instance.

Column	Datatype	Description
INSTANCE_NUMBER	NUMBER	Instance number used for instance registration (corresponds to the INSTANCE_NUMBER initialization parameter) See Also: "INSTANCE_NUMBER" on page 1-61
INSTANCE_NAME	VARCHAR2(16)	Name of the instance
HOST_NAME	VARCHAR2(64)	Name of the host machine
VERSION	VARCHAR2(17)	Database version
STARTUP_TIME	DATE	Time when the instance was started
STATUS	VARCHAR2(12)	Status of the instance: <ul style="list-style-type: none"> ■ STARTED - After STARTUP NOMOUNT ■ MOUNTED - After STARTUP MOUNT or ALTER DATABASE CLOSE ■ OPEN - After STARTUP or ALTER DATABASE OPEN ■ OPEN MIGRATE - After ALTER DATABASE OPEN { UPGRADE DOWNGRADE }
PARALLEL	VARCHAR2(3)	Indicates whether the instance is mounted in cluster database mode (YES) or not (NO)
THREAD#	NUMBER	Redo thread opened by the instance
ARCHIVER	VARCHAR2(7)	(STOPPED STARTED FAILED) FAILED means that the archiver failed to archive a log last time, but will try again within 5 minutes
LOG_SWITCH_WAIT	VARCHAR2(11)	The ARCHIVE LOG/CLEAR LOG/CHECKPOINT event log switching is waiting for. Note that if ALTER SYSTEM SWITCH LOGFILE is hung, but there is room in the current online redo log, then value is NULL
LOGINS	VARCHAR2(10)	ALLOWED RESTRICTED

Column	Datatype	Description
SHUTDOWN_PENDING	VARCHAR2(3)	YES NO
DATABASE_STATUS	VARCHAR2(17)	Status of the database
INSTANCE_ROLE	VARCHAR2(18)	Describes whether the instance is an active instance (PRIMARY_INSTANCE) or an inactive secondary instance (SECONDARY_INSTANCE), or UNKNOWN if the instance has been started but not mounted
ACTIVE_STATE	VARCHAR2(9)	(NORMAL QUIESCING QUIESCED). NORMAL indicates the database is in a normal state. QUIESCING indicates that the ALTER SYSTEM QUIESCE RESTRICTED statement has been issued: no new user transactions, queries, or PL/SQL statements are processed in this instance. User transactions, queries, or PL/SQL statements issued before the ALTER SYSTEM QUIESCE RESTRICTED statement are unaffected. DBA transactions, queries, or PL/SQL statements are also unaffected. QUIESCED indicates that the ALTER SYSTEM QUIESCE RESTRICTED statement has been issued: no user transactions, queries, or PL/SQL statements are processed. DBA transactions, queries, or PL/SQL statements are unaffected. User transactions, queries, or PL/SQL statements issued after the ALTER SYSTEM QUIESCE RESTRICTED statement are not processed. Note that a single ALTER SYSTEM QUIESCE RESTRICTED statement quiesces all instances in a Real Application Clusters environment. After this statement has been issued, some instances may enter into a quiesced state before other instances; the system is quiesced when all instances enter the quiesced state.

V\$INSTANCE_RECOVERY

V\$INSTANCE_RECOVERY monitors the mechanisms available to users to limit recovery I/O. Those mechanisms are:

- Set the LOG_CHECKPOINT_TIMEOUT initialization parameter
- Set the LOG_CHECKPOINT_INTERVAL initialization parameter
- Set the FAST_START_MTTR_TARGET initialization parameter
- Set the size of the smallest redo log

Column	Datatype	Description
RECOVERY_ESTIMATED_IOS	NUMBER	Number of dirty buffers in the buffer cache. In the Standard Edition, this column is always null.
ACTUAL_REDO_BLKs	NUMBER	Current actual number of redo blocks required for recovery
TARGET_REDO_BLKs	NUMBER	Current target number of redo blocks that must be processed for recovery. This value is the minimum value of the following 3 columns, and identifies which of the 3 user-defined limits determines checkpointing.
LOG_FILE_SIZE_REDO_BLKs	NUMBER	Maximum number of redo blocks required to guarantee that a log switch does not occur before the checkpoint completes.
LOG_CHKPT_TIMEOUT_REDO_BLKs	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_TIMEOUT parameter. The value displayed is not meaningful unless that parameter has been set.
LOG_CHKPT_INTERVAL_REDO_BLKs	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_INTERVAL parameter. The value displayed is not meaningful unless that parameter has been set.
FAST_START_IO_TARGET_REDO_BLKs	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always null.

Column	Datatype	Description
TARGET_MTTR	NUMBER	Effective MTTR (mean time to recover) target value in seconds. The TARGET_MTTR value is calculated based on the value of the FAST_START_MTTR_TARGET parameter (the TARGET_MTTR value is used internally), and is usually an approximation of the parameter's value. However, if the FAST_START_MTTR_TARGET parameter value is very small (for example, one second), or very large (for example, 3600 seconds), the calculation will produce a target value dictated by system limitations. In such cases, the TARGET_MTTR value will be the shortest calculated time, or the longest calculated time that recovery is expected to take. If FAST_START_MTTR_TARGET is not specified, the value of this field is the current estimated MTTR.
ESTIMATED_MTTR	NUMBER	Current estimated mean time to recover (MTTR) based on the number of dirty buffers and log blocks (0 if FAST_START_MTTR_TARGET is not specified). Basically, this value tells you how long you could expect recovery to take based on the work your system is doing right now.
CKPT_BLOCK_WRITES	NUMBER	Number of blocks written by checkpoint writes
OPTIMAL_LOGFILE_SIZE	NUMBER	Redo log file size (in megabytes) that is considered optimal based on the current setting of FAST_START_MTTR_TARGET. It is recommended that the user configure all online redo logs to be at least this value.
ESTD_CLUSTER_AVAILABLE_TIME	NUMBER	Estimated time (in seconds) that the cluster would become partially available should this instance fail. This column is only meaningful in a Real Application Clusters environment. In a non-RAC environment, the value of this column is null.
WRITES_MTTR	NUMBER	Number of writes driven by the FAST_START_MTTR_TARGET initialization parameter
WRITES_LOGFILE_SIZE	NUMBER	Number of writes driven by the smallest redo log file size
WRITES_LOG_CHECKPOINT_SETTINGS	NUMBER	Number of writes driven by the LOG_CHECKPOINT_INTERVAL or LOG_CHECKPOINT_TIMEOUT initialization parameter
WRITES_OTHER_SETTINGS	NUMBER	Number of writes driven by other reasons (such as the deprecated FAST_START_IO_TARGET initialization parameter)
WRITES_AUTOTUNE	NUMBER	Number of writes due to auto-tune checkpointing
WRITES_FULL_THREAD_CKPT	NUMBER	Number of writes due to full thread checkpoints

See Also:

- *Oracle Database Performance Tuning Guide* for more information on limiting recovery I/O, and how Oracle determines MTTR recovery times
- [LOG_CHECKPOINT_INTERVAL](#) on page 1-79
- [FAST_START_MTTR_TARGET](#) on page 1-54

V\$JAVA_LIBRARY_CACHE_MEMORY

V\$JAVA_LIBRARY_CACHE_MEMORY displays information about memory allocated to library cache memory objects in different namespaces for Java objects. A memory object is an internal grouping of memory for efficient management. A library cache object may consist of one or more memory objects.

Column	Datatype	Description
LC_NAMESPACE	VARCHAR2(15)	Library cache namespace
LC_INUSE_MEMORY_OBJECTS	NUMBER	Number of library cache memory objects currently in use in the Java pool

Column	Datatype	Description
LC_INUSE_MEMORY_SIZE	NUMBER	Total size of library cache in-use memory objects (in megabytes)
LC_FREEABLE_MEMORY_OBJECTS	NUMBER	Number of freeable library cache memory objects in the Java pool
LC_FREEABLE_MEMORY_SIZE	NUMBER	Size of library cache freeable memory objects (in megabytes)

V\$JAVA_POOL_ADVICE

V\$JAVA_POOL_ADVICE displays information about estimated parse time in the Java pool for different pool sizes. The sizes range from 10% of the current Java pool size or the amount of pinned Java library cache memory (whichever is higher) to 200% of the current Java pool size, in equal intervals. The value of the interval depends on the current size of the Java pool.

Parse time saved refers to the amount of time saved by keeping library cache memory objects in the Java pool, as opposed to having to reload these objects.

Column	Datatype	Description
JAVA_POOL_SIZE_FOR_ESTIMATE	NUMBER	Java pool size for the estimate (in megabytes)
JAVA_POOL_SIZE_FACTOR	NUMBER	Size factor with respect to the current Java pool size
ESTD_LC_SIZE	NUMBER	Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER	Estimated number of library cache memory objects in the Java pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER	Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a Java pool of the specified size. This is the time that would have been spent in reloading the required objects in the Java pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER	Estimated parse time saved factor with respect to the current Java pool size
ESTD_LC_LOAD_TIME	NUMBER	Estimated elapsed time (in seconds) for parsing in a Java pool of the specified size
ESTD_LC_LOAD_TIME_FACTOR	NUMBER	Estimated load time factor with respect to the current Java pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER	Estimated number of times a library cache memory object was found in a Java pool of the specified size

V\$LATCH

V\$LATCH shows aggregate latch statistics for both parent and child latches, grouped by latch name. Individual parent and child latch statistics are broken down in the views V\$LATCH_PARENT and V\$LATCH_CHILDREN.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of the latch object
LATCH#	NUMBER	Latch number
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(50)	Latch name
HASH	NUMBER	Latch hash
GETS	NUMBER	Number of times the latch was requested in willing-to-wait mode

Column	Datatype	Description
MISSES	NUMBER	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER	Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER	Number of times a no-wait latch request did not succeed (that is, missed)
WAITERS_WOKEN	NUMBER	For some latches, the session releasing the latch posts a session waiting for the latch. This counts the number of times a waiting session was awakened.
WAITS_HOLDING_LATCH	NUMBER	Number of waits for the latch while the waiter was holding a different latch
SPIN_GETS	NUMBER	Willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP[1 2 3]	NUMBER	Waits that slept 1 time through 3 times, respectively
SLEEP4	NUMBER	Waits that slept 4 or more times
SLEEP[5 6 7 8 9 10 11]	NUMBER	These columns are present for compatibility with previous releases of Oracle. No data is accumulated for these columns.
WAIT_TIME	NUMBER	Elapsed time spent waiting for the latch (in microseconds)

See Also:

- ["V\\$LATCH_CHILDREN"](#) on page 4-73
- ["V\\$LATCH_PARENT"](#) on page 4-74

V\$LATCH_CHILDREN

V\$LATCH_CHILDREN contains statistics about child latches. This view includes all columns of V\$LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of the latch object
LATCH#	NUMBER	Latch number of the parent latch
CHILD#	NUMBER	Child latch number (unique only to each parent latch)
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(50)	Latch name
HASH	NUMBER	Latch hash
GETS	NUMBER	Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER	Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER	Number of times a no-wait latch request did not succeed (that is, missed)
WAITERS_WOKEN	NUMBER	For some latches, the session releasing the latch posts a session waiting for the latch. This counts the number of times a waiting session was awakened.
WAITS_HOLDING_LATCH	NUMBER	Number of waits for the latch while the waiter was holding a different latch

Column	Datatype	Description
SPIN_GETS	NUMBER	Willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP[1 2 3]	NUMBER	Waits that slept 1 time through 3 times, respectively
SLEEP4	NUMBER	Waits that slept 4 or more times
SLEEP[5 6 7 8 9 10 11]	NUMBER	These columns are present for compatibility with previous releases of Oracle. No data is accumulated for these columns.
WAIT_TIME	NUMBER	Elapsed time spent waiting for the latch (in microseconds)

See Also: ["V\\$LATCH"](#) on page 4-72

V\$LATCH_MISSES

This view contains statistics about missed attempts to acquire a latch.

Column	Datatype	Description
PARENT_NAME	VARCHAR2(50)	Latch name of a parent latch
WHERE	VARCHAR2(64)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in LOCATION.
NWFAIL_COUNT	NUMBER	Number of times that no-wait acquisition of the latch failed
SLEEP_COUNT	NUMBER	Number of times that acquisition attempts caused sleeps
WTR_SLP_COUNT	NUMBER	Number of times a waiter slept from this loc
LONGHOLD_COUNT	NUMBER	Number of times someone held a latch for the entire duration of someone else's sleep
LOCATION	VARCHAR2(64)	Location that attempted to acquire the latch

V\$LATCH_PARENT

V\$LATCH_PARENT contains statistics about parent latches. The columns of V\$LATCH_PARENT are identical to those in V\$LATCH.

See Also: ["V\\$LATCH"](#) on page 4-72

V\$LATCHHOLDER

This view contains information about the current latch holders.

Column	Datatype	Description
PID	NUMBER	Identifier of the process holding the latch
SID	NUMBER	Identifier of the session that owns the latch
LADDR	RAW(4 8)	Latch address
NAME	VARCHAR2(64)	Name of the latch being held

V\$LATCHNAME

This view contains information about decoded latch names for the latches shown in V\$LATCH. The rows of V\$LATCHNAME have a one-to-one correspondence to the rows of V\$LATCH.

Column	Datatype	Description
LATCH#	NUMBER	Latch number
NAME	VARCHAR2(50)	Latch name
HASH	NUMBER	Latch hash

See Also: "[V\\$LATCH](#)" on page 4-72

V\$LIBRARY_CACHE_MEMORY

V\$LIBRARY_CACHE_MEMORY displays information about memory allocated to library cache memory objects in different namespaces. A memory object is an internal grouping of memory for efficient management. A library cache object may consist of one or more memory objects.

Column	Datatype	Description
LC_NAMESPACE	VARCHAR2(15)	Library cache namespace
LC_INUSE_MEMORY_OBJECTS	NUMBER	Number of library cache memory objects currently in use in the shared pool
LC_INUSE_MEMORY_SIZE	NUMBER	Total size of library cache in-use memory objects (in megabytes)
LC_FREEABLE_MEMORY_OBJECTS	NUMBER	Number of freeable library cache memory objects in the shared pool
LC_FREEABLE_MEMORY_SIZE	NUMBER	Size of library cache freeable memory objects (in megabytes)

V\$LIBRARYCACHE

This view contains statistics about library cache performance and activity.

Column	Datatype	Description
NAMESPACE	VARCHAR2(15)	Library cache namespace
GETS	NUMBER	Number of times a lock was requested for objects of this namespace
GETHITS	NUMBER	Number of times an object's handle was found in memory
GETHITRATIO	NUMBER	Ratio of GETHITS to GETS
PINS	NUMBER	Number of times a PIN was requested for objects of this namespace
PINHITS	NUMBER	Number of times all of the metadata pieces of the library object were found in memory
PINHITRATIO	NUMBER	Ratio of PINHITS to PINS
RELOADS	NUMBER	Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from disk
INVALIDATIONS	NUMBER	The total number of times objects in this namespace were marked invalid because a dependent object was modified
DLM_LOCK_REQUESTS	NUMBER	Number of GET requests lock instance locks
DLM_PIN_REQUESTS	NUMBER	Number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER	Number of release requests PIN instance locks
DLM_INVALIDATION_REQUESTS	NUMBER	Number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER	Number of invalidation pings received from other instances

V\$LICENSE

This view contains information about license limits.

Column	Datatype	Description
SESSIONS_MAX	NUMBER	Maximum number of concurrent user sessions allowed for the instance
SESSIONS_WARNING	NUMBER	Warning limit for concurrent user sessions for the instance
SESSIONS_CURRENT	NUMBER	Current number of concurrent user sessions
SESSIONS_HIGHWATER	NUMBER	Highest number of concurrent user sessions since the instance started
USERS_MAX	NUMBER	Maximum number of named users allowed for the database

V\$LOADISTAT

This view contains errors that occurred when updating indexes on a table during a load using the Direct Path API.

Column	Datatype	Description
OWNER	VARCHAR2(31)	Schema name
TABNAME	VARCHAR2(31)	Table name
INDEXNAME	VARCHAR2(31)	Index name
SUBNAME	VARCHAR2(31)	Index sub name
MESSAGE_NUM	NUMBER	Error message number
MESSAGE	VARCHAR2(4000)	Error message

V\$LOADPSTAT

This view contains statistics about the number of rows loaded into a partition, or subpartition, during a load using the Direct Path API.

Column	Datatype	Description
OWNER	VARCHAR2(31)	Schema name
TABNAME	VARCHAR2(31)	Table name
PARTNAME	VARCHAR2(31)	Partition name
LOADED	NUMBER	Number of rows loaded

V\$LOCK

This view lists the locks currently held by the Oracle Database and outstanding requests for a lock or latch.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of lock state object
KADDR	RAW(4 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock

Column	Datatype	Description
TYPE	VARCHAR2 (2)	Type of user or system lock The locks on the user types are obtained by user applications. Any process that is blocking others is likely to be holding one of these locks. The user type locks are: TM - DML enqueue TX - Transaction enqueue UL - User supplied The locks on the system types are held for extremely short periods of time. The system type locks are listed in Table 4-1 .
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: <ul style="list-style-type: none"> ▪ 0 - none ▪ 1 - null (NULL) ▪ 2 - row-S (SS) ▪ 3 - row-X (SX) ▪ 4 - share (S) ▪ 5 - S/Row-X (SSX) ▪ 6 - exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: <ul style="list-style-type: none"> ▪ 0 - none ▪ 1 - null (NULL) ▪ 2 - row-S (SS) ▪ 3 - row-X (SX) ▪ 4 - share (S) ▪ 5 - S/Row-X (SSX) ▪ 6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

Table 4-1 Values for the TYPE Column: System Types

System Type	Description	System Type	Description
BL	Buffer hash table instance	NA . . NZ	Library cache pin instance (A . . Z = namespace)
CF	Control file schema global enqueue	PF	Password File
CI	Cross-instance function invocation instance	PI , PS	Parallel operation
CU	Cursor bind	PR	Process startup
DF	Data file instance	QA . . QZ	Row cache instance (A . . Z = cache)
DL	Direct loader parallel index create	RT	Redo thread global enqueue
DM	Mount/startup db primary/secondary instance	SC	System change number instance
DR	Distributed recovery process	SM	SMON
DX	Distributed transaction entry	SN	Sequence number instance
FS	File set	SQ	Sequence number enqueue
HW	Space management operations on a specific segment	SS	Sort segment
IN	Instance number	ST	Space transaction enqueue
IR	Instance recovery serialization global enqueue	SV	Sequence number value

Table 4–1 (Cont.) Values for the TYPE Column: System Types

System Type	Description	System Type	Description
IS	Instance state	TA	Generic enqueue
IV	Library cache invalidation instance	TS	Temporary segment enqueue (ID2=0)
JQ	Job queue	TS	New block allocation enqueue (ID2=1)
KK	Thread kick	TT	Temporary table enqueue
LA . . LP	Library cache lock instance lock (A..P = namespace)	UN	User name
MM	Mount definition global enqueue	US	Undo segment DDL
MR	Media recovery	WL	Being-written redo log instance

V\$LOCK_ACTIVITY

V\$LOCK_ACTIVITY is deprecated.

Column	Datatype	Description
FROM_VAL	CHAR (4)	Global Cache Resource initial state; always NULL
TO_VAL	CHAR (1)	Global Cache Resource initial state; always S
ACTION_VAL	CHAR (21)	Description of the conversion; always Lock buffers for read
COUNTER	NUMBER	Number of times the lock operation executed

V\$LOCKED_OBJECT

This view lists all locks acquired by every transaction on the system.

Column	Datatype	Description
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
OBJECT_ID	NUMBER	Object ID being locked
SESSION_ID	NUMBER	Session ID
ORACLE_USERNAME	VARCHAR2 (30)	Oracle user name
OS_USER_NAME	VARCHAR2 (30)	OS user name
PROCESS	VARCHAR2 (12)	OS process ID
LOCKED_MODE	NUMBER	Lock mode

V\$LOG

V\$LOG displays log file information from the control file.

Column	Datatype	Description
GROUP#	NUMBER	Log group number
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
BYTES	NUMBER	Size of the log (in bytes)
MEMBERS	NUMBER	Number of members in the log group

Column	Datatype	Description
ARCHIVED	VARCHAR2 (3)	Archive status (YES or NO)
STATUS	VARCHAR2 (16)	Log status: <ul style="list-style-type: none"> ▪ UNUSED - Online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. ▪ CURRENT - Current redo log. This implies that the redo log is active. The redo log could be open or closed. ▪ ACTIVE - Log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It may or may not be archived. ▪ CLEARING - Log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE statement. After the log is cleared, the status changes to UNUSED. ▪ CLEARING_CURRENT - Current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. ▪ INACTIVE - Log is no longer needed for instance recovery. It may be in use for media recovery. It might or might not be archived. ▪ INVALIDATED - Archived the current redo log without a log switch.
FIRST_CHANGE#	NUMBER	Lowest system change number (SCN) in the log
FIRST_TIME	DATE	Time of the first SCN in the log

V\$LOG_HISTORY

V\$LOG_HISTORY displays log history information from the control file.

Column	Datatype	Description
RECID	NUMBER	Control file record ID
STAMP	NUMBER	Control file record stamp
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
FIRST_CHANGE#	NUMBER	Lowest system change number (SCN) in the log
FIRST_TIME	DATE	Time of the first entry (lowest SCN) in the log
NEXT_CHANGE#	NUMBER	Highest SCN in the log
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when the log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when the log was written

V\$LOGFILE

This view contains information about redo log files.

Column	Datatype	Description
GROUP#	NUMBER	Redo log group identifier number
STATUS	VARCHAR2 (7)	Status of the log member: <ul style="list-style-type: none"> ▪ INVALID - File is inaccessible ▪ STALE - File's contents are incomplete ▪ DELETED - File is no longer used ▪ null - File is in use

Column	Datatype	Description
TYPE	VARCHAR2(7)	Type of the logfile: <ul style="list-style-type: none"> ■ ONLINE ■ STANDBY
MEMBER	VARCHAR2(513)	Redo log member name
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)

V\$LOGHIST

This view contains log history information from the control file. This view is retained for historical compatibility. Oracle recommends that you use V\$LOG_HISTORY instead.

See Also: ["V\\$LOG_HISTORY"](#) on page 4-79

Column	Datatype	Description
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log
SWITCH_CHANGE#	NUMBER	SCN at which the log switch occurred; one more than highest SCN in the log

V\$LOGMNR_CONTENTS

This view contains log history information.

When a SELECT statement is executed against the V\$LOGMNR_CONTENTS view, the archive redo log files are read sequentially. Translated records from the redo log files are returned as rows in the V\$LOGMNR_CONTENTS view. This continues until either the filter criteria specified at startup (EndTime or endScn) are met or the end of the archive log file is reached.

Column	Datatype	Description
SCN	NUMBER	System change number (SCN) when the redo record was generated
CSCN	NUMBER	System change number (SCN) when the transaction committed; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation
TIMESTAMP	DATE	Timestamp when the redo record was generated
COMMIT_TIMESTAMP	DATE	Timestamp when the transaction committed; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation
THREAD#	NUMBER	Number of the thread which generated the redo record
LOG_ID	NUMBER	This column is deprecated.
XIDUSN	NUMBER	Transaction ID undo segment number of the transaction which generated the change
XIDSLT	NUMBER	Transaction ID slot number of the transaction which generated the change
XIDSQN	NUMBER	Transaction ID sequence number of the transaction which generated the change
PXIDUSN	NUMBER	Parent transaction ID undo segment number of a PDML transaction

Column	Datatype	Description
PXIDSLT	NUMBER	Parent transaction ID slot number of a PDML transaction
PXIDSQN	NUMBER	Parent transaction ID sequence number of a PDML transaction
RBASQN	NUMBER	RBA sequence number of the log that contained this redo record
RBABLK	NUMBER	RBA block number within the log file
RBABYTE	NUMBER	RBA byte offset within the block
UBAFIL	NUMBER	UBA file number identifying the file containing the undo block
UBABLK	NUMBER	UBA block number for the undo block
UBAREC	NUMBER	UBA record index within the undo block
UBASQN	NUMBER	UBA undo block sequence number
ABS_FILE#	NUMBER	Data block absolute file number of the block changed by the transaction
REL_FILE#	NUMBER	Data block relative file number. The file number is relative to the tablespace of the object.
DATA_BLK#	NUMBER	Data block number within the file
DATA_OBJ#	NUMBER	Data block object number identifying the object
DATA_OBJD#	NUMBER	Data block data object number identifying the object within the tablespace
SEG_OWNER	VARCHAR2(32)	Owner of the segment
SEG_NAME	VARCHAR2(256)	Name of the segment
TABLE_NAME	VARCHAR2(32)	Name of the table if the type is a table
SEG_TYPE	NUMBER	Segment type
SEG_TYPE_NAME	VARCHAR2(32)	Segment type name
TABLE_SPACE	VARCHAR2(32)	Tablespace name
ROW_ID	VARCHAR2(18)	Row ID
SESSION#	NUMBER	Session number of the session which generated the redo
SERIAL#	NUMBER	Serial number of the session which generated the redo
USERNAME	VARCHAR2(30)	Name of the user who executed the transaction
SESSION_INFO	VARCHAR2(4000)	Information about the database session that executed the transaction
TX_NAME	VARCHAR2(256)	Name of the transaction if it is a named transaction
ROLLBACK	NUMBER	Rollback request
OPERATION	VARCHAR2(32)	Operation
OPERATION_CODE	NUMBER	Operation code
SQL_REDO	VARCHAR2(4000)	SQL redo
SQL_UNDO	VARCHAR2(4000)	SQL undo
RS_ID	VARCHAR2(32)	Record set ID
SEQUENCE#	NUMBER	Sequence number
SSN	NUMBER	SQL sequence number
CSF	NUMBER	Continuation SQL flag
INFO	VARCHAR2(32)	Informational message
STATUS	NUMBER	Status
REDO_VALUE	NUMBER	Used as input to the DBMS_LOGMNR.MINE_VALUE() and DBMS_LOGMNR.COLUMN_PRESENT() functions
UNDO_VALUE	NUMBER	Used as input to the DBMS_LOGMNR.MINE_VALUE() and DBMS_LOGMNR.COLUMN_PRESENT() functions
SQL_COLUMN_TYPE	VARCHAR2(30)	This column is deprecated.
SQL_COLUMN_NAME	VARCHAR2(30)	This column is deprecated.

Column	Datatype	Description
REDO_LENGTH	NUMBER	This column is deprecated.
REDO_OFFSET	NUMBER	This column is deprecated.
UNDO_LENGTH	NUMBER	This column is deprecated.
UNDO_OFFSET	NUMBER	This column is deprecated.
DATA_OBJV#	NUMBER	Version number of the object DATA_OBJ#
SAFE_RESUME_SCN	NUMBER	Reserved for future use
XID	RAW(8)	Full transaction handle
PXID	RAW(8)	Full transaction handle of the parent transaction
AUDIT_SESSIONID	NUMBER	Audit session ID of the transaction

V\$LOGMNR_DICTIONARY

This view contains log history information.

Column	Datatype	Description
DB_NAME	VARCHAR2(9)	Name of the database
DB_ID	NUMBER	Database ID
DB_CREATED	DATE	Creation date of the source database (corresponds to the CREATED column in the V\$DATABASE view)
TIMESTAMP	DATE	Date when the dictionary was created
RESET_SCN	NUMBER	Reset log SCN when the dictionary was created
RESET_SCN_TIME	DATE	Timestamp of the reset log SCN when the dictionary was created
DB_VERSION_TIME	DATE	Version time for the source database (corresponds to the VERSION_TIME column in the V\$DATABASE view)
DB_CHARACTER_SET	VARCHAR2(30)	Character set of the source database
DB_VERSION	VARCHAR2(64)	This column is deprecated.
DB_STATUS	VARCHAR2(64)	This column is deprecated.
DICTIONARY_SCN	NUMBER	Database checkpoint SCN at which the dictionary was created
ENABLED_THREAD_MAP	RAW(16)	This column is deprecated.
DB_TXN_SCN	NUMBER	SCN at which the dictionary was created
FILENAME	VARCHAR2(512)	Dictionary file name
INFO	VARCHAR2(32)	Informational/Status message BAD_DATE indicates that the SCN of the dictionary file does not match the SCN range of the log files
STATUS	NUMBER	A NULL indicates a valid dictionary file for the list of log files. A non-NULL value indicates further information is contained in the INFO column as a text string.

V\$LOGMNR_LOGS

This view contains log information.

Column	Datatype	Description
LOG_ID	NUMBER	This column is deprecated.
FILENAME	VARCHAR2(512)	Name of the log file
LOW_TIME	DATE	Oldest date of any records in the file
HIGH_TIME	DATE	Most recent date of any records in the file

Column	Datatype	Description
DB_ID	NUMBER	Database ID
DB_NAME	VARCHAR2 (8)	Name of the database
RESET_SCN	NUMBER	Resetlogs SCN of the database incarnation that generated the log file
RESET_SCN_TIME	DATE	Resetlogs timestamp of the database incarnation that generated the log file
THREAD_ID	NUMBER	Thread number
THREAD_SEQN	NUMBER	Thread sequence number
LOW_SCN	NUMBER	SCN allocated when log switched into
NEXT_SCN	NUMBER	SCN after this log. Low SCN of the next log.
DICTIONARY_BEGIN	VARCHAR2 (3)	Indicates whether dictionary dumped to redo logs starts in this redo log (YES) or not (NO)
DICTIONARY_END	VARCHAR2 (3)	Indicates whether dictionary dumped to redo logs ends in this redo log (YES) or not (NO)
TYPE	VARCHAR2 (7)	Redo log file type: <ul style="list-style-type: none"> ■ ARCHIVED ■ ONLINE
BLOCKSIZE	NUMBER	Database block size
FILESIZE	NUMBER	Size of the redo file (in bytes)
INFO	VARCHAR2 (32)	Informational message. A value of MISSING_LOGFILE will be assigned to a row entry where a needed log file is missing from the list of log files.
STATUS	NUMBER	Status of the redo log file: <ul style="list-style-type: none"> ■ 0 - Will be read ■ 1 - First to be read ■ 2 - Not needed ■ 4 - Missing log file

V\$LOGMNR_PARAMETERS

This view contains log information.

Column	Datatype	Description
START_DATE	DATE	Date to start the search
REQUIRED_START_DATE	DATE	Required date to start the search if DDL tracking is enabled
END_DATE	DATE	Date to end the search
START_SCN	NUMBER	System change number to start the search
REQUIRED_START_SCN	NUMBER	Required system change number to start the search if DDL tracking is enabled
END_SCN	NUMBER	System change number to end the search
OPTIONS	NUMBER	Options specified for the current LogMiner session
INFO	VARCHAR2 (32)	This column is always null.
STATUS	NUMBER	This column is always 0.

V\$LOGSTDBY

V\$LOGSTDBY displays dynamic information about what is happening to the Data Guard log apply services. This view is very helpful when diagnosing performance problems during the logical application of archived redo logs to the standby database, and it can be helpful for other problems. This view is for logical standby databases only.

Column	Datatype	Description
SERIAL#	NUMBER	SQL Session serial number. This data is used when joining this view with the V\$SESSION and V\$PX_SESSION views.
LOGSTDBY_ID	NUMBER	Parallel query slave ID
PID	VARCHAR2(12)	Process ID of the SQL apply process
TYPE	VARCHAR2(30)	Indicates the task being performed by the process (COORDINATOR, APPLIER, ANALYZER, READER, PREPARER, or BUILDER)
STATUS_CODE	NUMBER	Status number (or Oracle error code) belonging to the STATUS message
STATUS	VARCHAR2(256)	Description of the current activity of the process
HIGH_SCN	NUMBER	Highest system change number (SCN) seen by the process. This column is used to confirm the progress of the individual process.

V\$LOGSTDBY_STATS

V\$LOGSTDBY_STATS displays LogMiner statistics, current state, and status information for the logical standby database during SQL apply. If SQL apply is not running, then the values for the statistics are cleared. This view is for logical standby databases only.

Column	Datatype	Description
NAME	VARCHAR2(64)	<p>Name of the statistic, state, or status:</p> <p>Note: Many of the following statistics are subject to change or deletion; programmers should write application code to tolerate missing or extra statistics.</p> <ul style="list-style-type: none"> ▪ Number of preparers ▪ Number of appliers ▪ Maximum SGA for LCR cache ▪ Parallel servers in use ▪ Transaction consistency ▪ Coordinator state ▪ Transactions scheduled ▪ Transactions applied ▪ Preparer memory alloc failures ▪ Builder memory alloc failures ▪ Attempts to handle low memory ▪ Successful low memory recovery ▪ Memory spills avoided ▪ Rollback attempts ▪ Successful rollbacks ▪ Memory spill attempts ▪ Successful memory spills ▪ Preparer ignored memory LWM ▪ Builder ignored memory LWM ▪ Mining resumed ▪ Number of DDL transactions applied ▪ Number of DDL transactions skipped ▪ Number of table creation and index changes ▪ Which table or tables are experiencing the most activity from SQL apply ▪ Average apply rate for the SQL apply
VALUE	VARCHAR2(64)	Value of the statistic or state information

V\$MANAGED_STANDBY

V\$MANAGED_STANDBY displays current status information for some Oracle Database processes related to physical standby databases in the Data Guard environment. This view does not persist after an instance shutdown.

Column	Datatype	Description
PROCESS	VARCHAR2(9)	Type of process whose information is being reported: <ul style="list-style-type: none"> ▪ RFS - Remote file server ▪ MRP0 - Detached recovery server process ▪ MR(fg) - Foreground recovery session ▪ ARCH - Archiver process ▪ FGRD ▪ LGWR ▪ RFS(FAL) ▪ RFS(NEXP)
PID	NUMBER	Operating system process identifier of process
STATUS	VARCHAR2(12)	Current process status: <ul style="list-style-type: none"> ▪ UNUSED - No active process ▪ ALLOCATED - Process is active but not currently connected to a primary database ▪ CONNECTED - Network connection established to a primary database ▪ ATTACHED - Process is actively attached and communicating to a primary database ▪ IDLE - Process is not performing any activities ▪ ERROR - Process has failed ▪ OPENING - Process is opening the archived redo log ▪ CLOSING - Process has completed archival and is closing the archived redo log ▪ WRITING - Process is actively writing redo data to the archived redo log ▪ RECEIVING - Process is receiving network communication ▪ ANNOUNCING - Process is announcing the existence of a potential dependent archived redo log ▪ REGISTERING - Process is registering the existence of a completed dependent archived redo log ▪ WAIT_FOR_LOG - Process is waiting for the archived redo log to be completed ▪ WAIT_FOR_GAP - Process is waiting for the archive gap to be resolved ▪ APPLYING_LOG - Process is actively applying the archived redo log to the standby database
CLIENT_PROCESS	VARCHAR2(8)	Identifies the corresponding primary database process: <ul style="list-style-type: none"> ▪ Archival - Foreground (manual) archival process (SQL) ▪ ARCH - Background ARCn process ▪ LGWR - Background LGWR process
CLIENT_PID	VARCHAR2(40)	Operating system process identifier of the client process
CLIENT_DBID	VARCHAR2(40)	Database identifier of the primary database
GROUP#	VARCHAR2(40)	Standby redo log group
RESETLOG_ID	NUMBER	Resetlogs identifier of the archived redo log
THREAD#	NUMBER	Archived redo log thread number
SEQUENCE#	NUMBER	Archived redo log sequence number

V\$MAP_COMP_LIST

Column	Datatype	Description
BLOCK#	NUMBER	Last processed archived redo log block number
BLOCKS	NUMBER	Size of the archived redo log in 512-byte blocks
DELAY_MINS	NUMBER	Archived redo log delay interval in minutes
KNOWN_AGENTS	NUMBER	Total number of standby database agents processing an archived redo log
ACTIVE_AGENTS	NUMBER	Number of standby database agents actively processing an archived redo log

V\$MAP_COMP_LIST

V\$MAP_COMP_LIST contains supplementary information for all element mapping structures.

Column	Datatype	Description
ELEM_IDX	NUMBER	Index corresponding to element
NUM_COMP	NUMBER	Number of components (maximum is 5)
COMP1_NAME	VARCHAR2(256)	Name of the first component
COMP1_VAL	VARCHAR2(256)	Value of the first component
COMP2_NAME	VARCHAR2(256)	Name of the second component
COMP2_VAL	VARCHAR2(256)	Value of the second component
COMP3_NAME	VARCHAR2(256)	Name of the third component
COMP3_VAL	VARCHAR2(256)	Value of the third component
COMP4_NAME	VARCHAR2(256)	Name of the fourth component
COMP4_VAL	VARCHAR2(256)	Value of the fourth component
COMP5_NAME	VARCHAR2(256)	Name of the fifth component
COMP5_VAL	VARCHAR2(256)	Value of the fifth component

V\$MAP_ELEMENT

V\$MAP_ELEMENT contains a list of all element mapping structures in the SGA of the instance.

Column	Datatype	Description
ELEM_NAME	VARCHAR2(256)	Element name
ELEM_IDX	NUMBER	Index corresponding to element
ELEM_CFGID	VARCHAR2(256)	Configuration ID: N/A if configuration ID is not supported
ELEM_TYPE	VARCHAR2(12)	Element type (MIRROR, STRIPE, RAID5, CONCATENATED, PARTITION, DISK, or NONE)
ELEM_SIZE	NUMBER	Element Size in HKB
ELEM_NSUBELEM	NUMBER	Number of Subelements
ELEM_DESCR	VARCHAR2(256)	Element Description
STRIPE_SIZE	NUMBER	Stripe Size in HKB for RAID-5 and STRIPE elements, 0 for the remaining types
LIB_IDX	NUMBER	Index of the library which claims ownership of the element

V\$MAP_EXT_ELEMENT

V\$MAP_EXT_ELEMENT contains supplementary information for all element mapping structures.

Column	Datatype	Description
ELEM_IDX	NUMBER	Index corresponding to element
NUM_ATTRB	NUMBER	Number of Attributes (maximum is 5)
ATTRB1_NAME	VARCHAR2(256)	Name of the first Attribute
ATTRB1_VAL	VARCHAR2(256)	Value of the first attribute
ATTRB2_NAME	VARCHAR2(256)	Name of the second attribute
ATTRB2_VAL	VARCHAR2(256)	Value of the second attribute
ATTRB3_NAME	VARCHAR2(256)	Name of the third attribute
ATTRB3_VAL	VARCHAR2(256)	Value of the third attribute
ATTRB4_NAME	VARCHAR2(256)	Name of the fourth attribute
ATTRB4_VAL	VARCHAR2(256)	Value of the fourth attribute
ATTRB5_NAME	VARCHAR2(256)	Name of the fifth attribute
ATTRB5_VAL	VARCHAR2(256)	Value of the fifth attribute

V\$MAP_FILE

V\$MAP_FILE contains a list of all file mapping structures in the shared memory of the instance.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	Index corresponding to file
FILE_CFGID	VARCHAR2(256)	Configuration ID: N/A if configuration ID is not supported
FILE_STATUS	VARCHAR2(7)	Status of the mapping information: <ul style="list-style-type: none"> ■ VALID - file mapping information is latest ■ INVALID - mapping needs to be refreshed
FILE_NAME	VARCHAR2(256)	Absolute file name
FILE_TYPE	VARCHAR2(11)	File type (DATAFILE, SPFILE, TEMPFILE, CONTROLFILE, LOGFILE, or ARCHIVEFILE)
FILE_STRUCTURE	VARCHAR2(9)	File structure (FILE, RAWVOLUME, RAWDEVICE, or NONE)
FILE_SIZE	NUMBER	File size in HKB (Half KB)
FILE_NEXTS	NUMBER	Number of file extents in the file (not necessarily the same as the number of file extents mapped)
LIB_IDX	NUMBER	Index of mapping library claiming ownership of the file

V\$MAP_FILE_EXTENT

V\$MAP_FILE_EXTENT contains a list of all file extent mapping structures in the shared memory of the instance.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	File index (corresponds to FILE_MAP_IDX in V\$MAP_FILE)
EXT_NUM	NUMBER	File extent number
EXT_ELEM_OFF	NUMBER	Element offset in HKB

Column	Datatype	Description
EXT_SIZE	NUMBER	File extent size in HKB
EXT_FILE_OFF	NUMBER	File Offset in HKB
EXT_TYPE	VARCHAR2(6)	File Extent Type (DATA, PARITY, or NONE)
ELEM_IDX	NUMBER	Index in V\$MAP_ELEMENT corresponding to the element where the file extent resides

V\$MAP_FILE_IO_STACK

V\$MAP_FILE_IO_STACK displays the hierarchical arrangement of storage containers for files. Each row in the view represents a level in the hierarchy.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	File index (corresponds to FILE_MAP_IDX in V\$MAP_FILE)
DEPTH	NUMBER	Element depth within the I/O stack
ELEM_IDX	NUMBER	Index corresponding to element
CU_SIZE	NUMBER	Contiguous set of logical blocks of the file, in HKB units, that is resident contiguously on the element
STRIDE	NUMBER	Number of HKB between contiguous units (CU) in the file that are contiguous on this element. Used in RAID5 and striped files.
NUM_CU	NUMBER	Number of contiguous units that are adjacent to each other on this element that are separated by STRIDE HKB in the file. In RAID5, the number of contiguous units also include the parity stripes.
ELEM_OFFSET	NUMBER	Element offset in HKB units
FILE_OFFSET	NUMBER	Offset in HKB units from the start of the file to the first byte of the contiguous units
DATA_TYPE	VARCHAR2(15)	Datatype (DATA, PARITY, or DATA AND PARITY)
PARITY_POS	NUMBER	Position of the parity. Only for RAID5. This field is needed to distinguish the parity from the data part.
PARITY_PERIOD	NUMBER	Parity period. Only for RAID5.
ID	NUMBER	Unique identifier
PARENT_ID	NUMBER	Parent identifier

V\$MAP_LIBRARY

V\$MAP_LIBRARY contains a list of all mapping libraries dynamically loaded by the external process.

Column	Datatype	Description
LIB_IDX	NUMBER	Index corresponding to library
LIB_NAME	VARCHAR2(256)	Absolute library name
VENDOR_NAME	VARCHAR2(64)	Name of the vendor implementing the library
PROTOCOL_NUM	NUMBER	Mapping protocol that the library supports
VERSION_NUM	VARCHAR2(32)	Version number
PATH_NAME	VARCHAR2(1024)	Path name
MAP_FILE	VARCHAR2(1)	Indicates whether this library supports mapping files (Y) or not (N)

Column	Datatype	Description
FILE_CFGID	VARCHAR2(13)	Type of configuration ID supported for files: <ul style="list-style-type: none"> ■ NONE - not supported ■ PERSISTENT ■ NONPERSISTENT
MAP_ELEM	VARCHAR2(1)	Indicates whether this library supports mapping elements (Y) or not (N)
ELEM_CFGID	VARCHAR2(13)	Type of configuration id supported for elements: <ul style="list-style-type: none"> ■ NONE - not supported ■ PERSISTENT ■ NONPERSISTENT
MAP_SYNC	VARCHAR2(1)	Indicates whether this library needs to be explicitly synced so that future mappings reflect the most recent changes (Y) or not (N). Note that configuration IDs cannot be supported if the library needs to be explicitly synced.

V\$MAP_SUBELEMENT

V\$MAP_SUBELEMENT contains a list of all subelement mapping structures in the shared memory of the instance.

Column	Datatype	Description
CHILD_IDX	NUMBER	Index in V\$MAP_ELEMENT corresponding to child element
PARENT_IDX	NUMBER	Index in V\$MAP_ELEMENT corresponding to parent element
SUB_NUM	NUMBER	Subelement number
SUB_SIZE	NUMBER	Subelement size in HKB
ELEM_OFFSET	NUMBER	Offset in HKB on child element
SUB_FLAGS	NUMBER	Subelement flags (currently unused)

V\$METRICNAME

V\$METRICNAME displays the mapping of the name of metrics to their metric ID.

Column	Datatype	Description
GROUP_ID	NUMBER	Metric group ID
GROUP_NAME	VARCHAR2(64)	Metric group name
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric name
METRIC_UNIT	VARCHAR2(64)	Unit of measurement

V\$MTTR_TARGET_ADVICE

V\$MTTR_TARGET_ADVICE contains rows that predict the number of physical I/Os for the MTTR corresponding to each row. The rows also compute a physical I/O factor, which is the ratio of the number of estimated I/Os to the number of I/Os actually performed by the current MTTR setting during the measurement interval.

The content of the view is empty if MTTR advisory has not been turned on since database startup. Otherwise, it returns the advisory information collected. If advisory is currently off, then this information comes from the last time MTTR advisory was on. FAST_START_MTTR_TARGET must be set to a nonzero value if the STATISTICS_LEVEL parameter is dynamically modified to turn MTTR advisory on.

If the `FAST_START_MTTR_TARGET` parameter is changed while MTTR advisory is on, then MTTR advisory is temporarily turned off until the new `FAST_START_MTTR_TARGET` setting takes effect. During this transition period, the contents of `V$MTTR_TARGET_ADVICE` reflect the simulation result for the old MTTR setting.

Column	Datatype	Description
<code>MTTR_TARGET_FOR_ESTIMATE</code>	NUMBER	MTTR setting being simulated. Equal to the current MTTR setting if this is the first row of the view.
<code>ADVICE_STATUS</code>	VARCHAR2(5)	Current status of MTTR simulation (ON, READY, or OFF)
<code>DIRTY_LIMIT</code>	NUMBER	Dirty buffer limit derived from the MTTR being simulated
<code>ESTD_CACHE_WRITES</code>	NUMBER	Estimated number of cache physical writes under this MTTR
<code>ESTD_CACHE_WRITE_FACTOR</code>	NUMBER	Estimated cache physical write ratio under this MTTR. It is the ratio of the estimated number of cache writes to the number of cache writes under the current MTTR setting.
<code>ESTD_TOTAL_WRITES</code>	NUMBER	Estimated total number of physical writes under this MTTR
<code>ESTD_TOTAL_WRITE_FACTOR</code>	NUMBER	Estimated total physical write ratio under this MTTR. It is the ratio of the estimated total number of physical writes to the total number of physical writes under the current MTTR setting.
<code>ESTD_TOTAL_IOS</code>	NUMBER	Estimated total number of I/Os under this MTTR
<code>ESTD_TOTAL_IO_FACTOR</code>	NUMBER	Estimated total I/O ratio under this MTTR. It is the ratio of the estimated total number of I/Os to the total number of I/Os under the current MTTR setting.

V\$MVREFRESH

`V$MVREFRESH` displays information about the materialized views currently being refreshed.

Column	Datatype	Description
<code>SID</code>	NUMBER	Session identifier
<code>SERIAL#</code>	NUMBER	Session serial number, which is used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends with, and another session begins with, the same session ID.
<code>CURRMVOWNER</code>	VARCHAR2(31)	Owner of the materialized view currently being refreshed. The materialized view resides in this user's schema.
<code>CURRMVNAME</code>	VARCHAR2(31)	Name of the materialized view currently being refreshed

V\$MYSTAT

This view contains statistics on the current session.

Column	Datatype	Description
<code>SID</code>	NUMBER	ID of the current session
<code>STATISTIC#</code>	NUMBER	Number of the statistic
<code>VALUE</code>	NUMBER	Value of the statistic

Dynamic Performance (V\$) Views: V\$NLS_PARAMETERS to V\$WAITSTAT

This chapter contains the dynamic performance views V\$NLS_PARAMETERS to V\$WAITSTAT.

V\$NLS_PARAMETERS

This view contains current values of NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Parameter name: NLS_CALENDAR, NLS_CHARACTERSET, NLS_CURRENCY, NLS_DATE_FORMAT, NLS_DATE_LANGUAGE, NLS_ISO_CURRENCY, NLS_LANGUAGE, NLS_NUMERIC_CHARACTERS, NLS_SORT, NLS_TERRITORY, NLS_UNION_CURRENCY, NLS_NCHAR_CHARACTERSET, NLS_COMP
VALUE	VARCHAR2(64)	NLS parameter value

V\$NLS_VALID_VALUES

This view lists all valid values for NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Parameter name (LANGUAGE SORT TERRITORY CHARACTERSET)
VALUE	VARCHAR2(64)	NLS parameter value

V\$OBJECT_DEPENDENCY

This view can be used to determine what objects are depended on by a package, procedure, or cursor that is currently loaded in the shared pool. For example, together with V\$SESSION and V\$SQL, it can be used to determine which tables are used in the SQL statement that a user is currently executing.

See Also: "V\$SESSION" on page 5-37 and "V\$SQL" on page 5-53

Column	Datatype	Description
FROM_ADDRESS	RAW(4 8)	Address of a procedure, package, or cursor that is currently loaded in the shared pool

V\$OBJECT_USAGE

Column	Datatype	Description
FROM_HASH	NUMBER	Hash value of a procedure, package, or cursor that is currently loaded in the shared pool
TO_OWNER	VARCHAR2(64)	Owner of the object that is depended on
TO_NAME	VARCHAR2(1000)	Name of the object that is depended on
TO_ADDRESS	RAW(4 8)	Address of the object that is depended on. These can be used to look up more information on the object in V\$DB_OBJECT_CACHE.
TO_HASH	NUMBER	Hash value of the object that is depended on. These can be used to look up more information on the object in V\$DB_OBJECT_CACHE.
TO_TYPE	NUMBER	Type of the object that is depended on

V\$OBJECT_USAGE

You can use this view to monitor index usage. The view displays statistics about index usage gathered from the database. All indexes that have been used at least once can be monitored and displayed in this view.

Column	Datatype	Description
INDEX_NAME	VARCHAR2(30)	Index name in sys.obj\$.name
TABLE_NAME	VARCHAR2(30)	Table name in sys.obj\$.name
MONITORING	VARCHAR2(3)	YES NO
USED	VARCHAR2(3)	YES NO
START_MONITORING	VARCHAR2(19)	Start monitoring time in sys.object_stats.start_monitoring
END_MONITORING	VARCHAR2(19)	End monitoring time in sys.object_stats.end_monitoring

V\$OBSOLETE_BACKUP_FILES

V\$OBSOLETE_BACKUP_FILES displays all obsolete backups, copies, and archived logs according to the current retention policy. This view requires that the database is set using the DBMS_RCVMAN.SETDATABASE procedure.

Column	Datatype	Description
PKEY	NUMBER	Primary key for the backup
BACKUP_TYPE	VARCHAR2(32)	Type of the backup: <ul style="list-style-type: none">■ BACKUP SET■ COPY■ PROXY COPY
FILE_TYPE	VARCHAR2(32)	Type of the file: <ul style="list-style-type: none">■ DATAFILE■ CONTROLFILE■ SPFILE■ REDO LOG■ PIECE
KEEP	VARCHAR2(3)	Indicates whether the backup has a retention policy different from the value for CONFIGURE RETENTION POLICY (YES) or not (NO)
KEEP_UNTIL	DATE	If the KEEP UNTIL TIME clause of the BACKUP command was specified, then this column shows the date after which the backup becomes obsolete. If the column is null and KEEP_OPTIONS is not null, the backup never becomes obsolete.

Column	Datatype	Description
KEEP_OPTIONS	VARCHAR2(13)	KEEP options for the backup: <ul style="list-style-type: none"> LOGS - RMAN keeps the logs needed to recover the backup NOLOGS - RMAN does not keep the logs needed to recover the backup If this column is null, then the backup has no KEEP options and will be made obsolete based on the retention policy.
STATUS	VARCHAR2(16)	Status of the backup: <ul style="list-style-type: none"> AVAILABLE UNAVAILABLE EXPIRED OTHER
FNAME	VARCHAR2(1024)	Name of the file
TAG	VARCHAR2(32)	Tag of the piece, copy, or proxy copy
MEDIA	VARCHAR2(80)	Media ID of the piece or proxy copy
RECID	NUMBER	Recid of the record in the controlfile
STAMP	NUMBER	Stamp of the record in the controlfile
DEVICE_TYPE	VARCHAR2(255)	Type of media device that stores the backup
BLOCK_SIZE	NUMBER	Block size for the backup (in bytes)
COMPLETION_TIME	DATE	Time when the backup completed
BS_KEY	NUMBER	Primary key of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_COUNT	NUMBER	Count of the backup set from the controlfile record (valid only when BACKUP_TYPE is BACKUP SET)
BS_STAMP	NUMBER	Stamp of the backup set from the controlfile record (valid only when BACKUP_TYPE is BACKUP SET)
BS_TYPE	VARCHAR2(32)	Type of the backup set (valid only when BACKUP_TYPE is BACKUP SET): <ul style="list-style-type: none"> DATAFILE ARCHIVED LOG
BS_INCR_TYPE	VARCHAR2(32)	Incremental level of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_PIECES	NUMBER	Number of backup pieces in the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_COMPLETION_TIME	DATE	Completion time of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BP_PIECE#	NUMBER	Number of the backup piece (valid only when FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BP_COPY#	NUMBER	Copy number of the backup piece (valid only when FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
DF_FILE#	NUMBER	Absolute file number of the datafile (valid only when FILE_TYPE is DATAFILE)
DF_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the control file or datafile was created (valid only when FILE_TYPE is DATAFILE)
DF_CREATION_CHANGE#	NUMBER	Creation SCN of the control file or datafile (valid only when FILE_TYPE is DATAFILE)
DF_CHECKPOINT_CHANGE#	NUMBER	System change number (SCN) of the most recent control file or datafile checkpoint (valid only when FILE_TYPE is DATAFILE)
DF_CKP_MOD_TIME	DATE	Modification time in case of SPFILE, otherwise time when the control file or datafile was checkpointed (valid only when FILE_TYPE is DATAFILE)
RL_THREAD#	NUMBER	Number of the redo thread (valid only when FILE_TYPE is REDO LOG)
RL_SEQUENCE#	NUMBER	Log sequence number (valid only when FILE_TYPE is REDO LOG)

V\$OBSOLETE_PARAMETER

Column	Datatype	Description
RL_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the record was created (valid only when FILE_TYPE is REDO LOG)
RL_FIRST_CHANGE#	NUMBER	First SCN of the redo log (valid only when FILE_TYPE is REDO LOG)
RL_FIRST_TIME	DATE	Time when Oracle switched into the redo log (valid only when FILE_TYPE is REDO LOG)
RL_NEXT_CHANGE#	NUMBER	First SCN of the next redo log in the thread (valid only when FILE_TYPE is REDO LOG)
RL_NEXT_TIME	DATE	First timestamp of the next redo log in the thread (valid only when FILE_TYPE is REDO LOG)

V\$OBSOLETE_PARAMETER

V\$OBSOLETE_PARAMETER displays information about obsolete initialization parameters. If any row of the view contains TRUE in the ISSPECIFIED column, then you should examine why.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the parameter
ISSPECIFIED	VARCHAR2(5)	Indicates whether the parameter was specified in the parameter file (TRUE) or not (FALSE)

V\$OFFLINE_RANGE

This view displays datafile offline information from the control file. Note that the last offline range of each datafile is kept in the DATAFILE record.

An offline range is created for a datafile when its tablespace is first altered to be OFFLINE NORMAL or READ ONLY, and then subsequently altered to be ONLINE or read/write. Note that no offline range is created if the datafile itself is altered to be OFFLINE or if the tablespace is altered to be OFFLINE IMMEDIATE.

See Also: ["V\\$DATAFILE"](#) on page 4-44

Column	Datatype	Description
RECID	NUMBER	Record ID
STAMP	NUMBER	Record stamp
FILE#	NUMBER	Datafile number
OFFLINE_CHANGE#	NUMBER	SCN at which offlined
ONLINE_CHANGE#	NUMBER	SCN at which onlined
ONLINE_TIME	DATE	Time of offline SCN
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the record
RESETLOGS_TIME	DATE	Resetlogs timestamp of the record

V\$OPEN_CURSOR

This view lists cursors that each user session currently has opened and parsed.

Column	Datatype	Description
SADDR	RAW(4 8)	Session address
SID	NUMBER	Session identifier

Column	Datatype	Description
USER_NAME	VARCHAR2 (30)	User that is logged in to the session
ADDRESS	RAW (4 8)	Used with HASH_VALUE to uniquely identify the SQL statement being executed in the session
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify the SQL statement being executed in the session
SQL_ID	VARCHAR2 (13)	SQL identifier of the SQL statement being executed in the session
SQL_TEXT	VARCHAR2 (60)	First 60 characters of the SQL statement that is parsed into the open cursor

V\$OPTION

This view lists options that are installed with the Oracle Database.

Column	Datatype	Description
PARAMETER	VARCHAR2 (64)	Name of the option
VALUE	VARCHAR2 (64)	TRUE if the option is installed

V\$OSSTAT

V\$OSSTAT displays system utilization statistics from the operating system. One row is returned for each system statistic.

Column	Datatype	Description
STAT_NAME	VARCHAR2 (64)	Name of the statistic (see Table 5-1)
VALUE	NUMBER	Instantaneous statistic value
OSSTAT_ID	NUMBER	Statistic ID

Table 5-1 V\$OSSTAT Statistics

Statistic Name	Description
NUM_CPUS	Number of CPUs or processors available
IDLE_TICKS	Number of hundredths of a second that a processor has been idle, totalled over all processors
BUSY_TICKS	Number of hundredths of a second that a processor has been busy executing user or kernel code, totalled over all processors
USER_TICKS	Number of hundredths of a second that a processor has been busy executing user code, totalled over all processors
SYS_TICKS	Number of hundredths of a second that a processor has been busy executing kernel code, totalled over all processors
IOWAIT_TICKS	Number of hundredths of a second that a processor has been waiting for I/O to complete, totalled over all processors
NICE_TICKS	Number of hundredths of a second that a processor has been busy executing low-priority user code, totalled over all processors
AVG_IDLE_TICKS	Number of hundredths of a second that a processor has been idle, averaged over all processors
AVG_BUSY_TICKS	Number of hundredths of a second that a processor has been busy executing user or kernel code, averaged over all processors
AVG_USER_TICKS	Number of hundredths of a second that a processor has been busy executing user code, averaged over all processors
AVG_SYS_TICKS	Number of hundredths of a second that a processor has been busy executing kernel code, averaged over all processors

Table 5–1 (Cont.) V\$OSSTAT Statistics

Statistic Name	Description
AVG_IOWAIT_TICKS	Number of hundredths of a second that a processor has been waiting for I/O to complete, averaged over all processors
AVG_NICE_TICKS	Number of hundredths of a second that a processor has been busy executing low-priority user code, averaged over all processors
OS_CPU_WAIT_TIME	Total number of hundredths of a second that processes have been in a ready state, waiting to be selected by the operating system scheduler to run
RSRC_MGR_CPU_WAIT_TIME	Total number of hundredths of a second that Oracle processes have been in a ready state, waiting for CPU to be available for their consumer group in the currently active resource plan
IN_BYTES	Total number of bytes that have been paged in
OUT_BYTES	Total number of bytes that have been paged out
FS_IN_BYTES	Total number of bytes that have been paged in due to the file system
FS_OUT_BYTES	Total number of bytes that have been paged out due to the file system
AVG_IN_BYTES	Number of bytes that have been paged in, averaged over all processors
AVG_OUT_BYTES	Total number of bytes that have been paged out, averaged over all processors
AVG_FS_IN_BYTES	Total number of bytes that have been paged in due to the file system, averaged over all processors
AVG_FS_OUT_BYTES	Total number of bytes that have been paged out due to the file system, averaged over all processors

Note: The availability of all statistics except for NUM_CPUS and RSRC_MGR_CPU_WAIT_TIME is subject to the operating system platform on which the Oracle Database is running. DBMS_DEFER_SYS.PUSH function for a push to use parallel propagation. Otherwise, the push uses serial propagation, and no data appears in this view during the push.

V\$PARALLEL_DEGREE_LIMIT_MTH

This view displays all available parallel degree limit resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the parallel degree limit resource allocation method

V\$PARAMETER

V\$PARAMETER displays information about the initialization parameters that are currently in effect for the session. A new session inherits parameter values from the instance-wide values displayed by the V\$SYSTEM_PARAMETER view.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter

Column	Datatype	Description
TYPE	NUMBER	Parameter type: <ul style="list-style-type: none"> ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value
DISPLAY_VALUE	VARCHAR2(512)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(9)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: <ul style="list-style-type: none"> ■ IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEM, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: <ul style="list-style-type: none"> ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update
HASH	NUMBER	Hash value for the parameter name

V\$PARAMETER2

V\$PARAMETER2 displays information about the initialization parameters that are currently in effect for the session, with each list parameter value appearing as a row in the view. A new session inherits parameter values from the instance-wide values displayed in the V\$SYSTEM_PARAMETER2 view.

Presenting the list parameter values in this format enables you to quickly determine the values for a list parameter. For example, if a parameter value is a , b, then the V\$PARAMETER view does not tell you if the parameter has two values (both a and b) or one value (a , b). V\$PARAMETER2 makes the distinction between the list parameter values clear.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter
TYPE	NUMBER	Parameter type: <ul style="list-style-type: none"> ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value
DISPLAY_VALUE	VARCHAR2(512)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(6)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: <ul style="list-style-type: none"> ■ IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEM, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: <ul style="list-style-type: none"> ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
ORDINAL	NUMBER	Position (ordinal number) of the parameter value. Useful only for parameters whose values are lists of strings.

Column	Datatype	Description
UPDATE_COMMENT	VARCHAR2 (255)	Comments associated with the most recent update

V\$PGA_TARGET_ADVICE

V\$PGA_TARGET_ADVICE predicts how the cache hit percentage and over allocation count statistics displayed by the V\$PGASTAT performance view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. The prediction is performed for various values of the PGA_AGGREGATE_TARGET parameter, selected around its current value. The advice statistic is generated by simulating the past workload run by the instance.

The content of the view is empty if PGA_AGGREGATE_TARGET is not set. In addition, the content of this view is not updated if the STATISTICS_LEVEL parameter is set to BASIC . Base statistics for this view are reset at instance startup and when the value of the PGA_AGGREGATE_TARGET initialization parameter is dynamically modified.

Column	Datatype	Description
PGA_TARGET_FOR_ESTIMATE	NUMBER	Value of PGA_AGGREGATE_TARGET for this prediction (in bytes)
PGA_TARGET_FACTOR	NUMBER	PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2 (3)	Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
BYTES_PROCESSED	NUMBER	Total bytes processed by all the work areas considered by this advice (in bytes)
ESTD_EXTRA_BYTES_RW	NUMBER	Estimated number of extra bytes which would be read or written if PGA_AGGREGATE_TARGET was set to the value of the PGA_TARGET_FOR_ESTIMATE column. This number is derived from the estimated number and size of work areas which would run in one-pass (or multi-pass) for that value of PGA_AGGREGATE_TARGET.
ESTD_PGA_CACHE_HIT_PERCENTAGE	NUMBER	Estimated value of the cache hit percentage statistic when PGA_AGGREGATE_TARGET equals PGA_TARGET_FOR_ESTIMATE. This column is derived from the above two columns and is equal to BYTES_PROCESSED / (BYTES_PROCESSED + ESTD_EXTRA_BYTES_RW)
ESTD_OVERALLOC_COUNT	NUMBER	Estimated number of PGA memory over-allocations if the value of PGA_AGGREGATE_TARGET is set to PGA_TARGET_FOR_ESTIMATE. A nonzero value means that PGA_TARGET_FOR_ESTIMATE is not large enough to run the work area workload. Hence, the DBA should not set PGA_AGGREGATE_TARGET to PGA_TARGET_FOR_ESTIMATE since Oracle will not be able to honor that target.

See Also: *Oracle Database Performance Tuning Guide* for information on tuning the PGA_AGGREGATE_TARGET initialization parameter using the PGA advice views

V\$PGA_TARGET_ADVICE_HISTOGRAM

V\$PGA_TARGET_ADVICE_HISTOGRAM predicts how statistics displayed by the V\$SQL_WORKAREA_HISTOGRAM dynamic view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. This prediction is performed for various values of the PGA_AGGREGATE_TARGET parameter, selected around its current value. The advice statistic is generated by simulating the past workload run by the instance.

The content of the view is empty if PGA_AGGREGATE_TARGET is not set. In addition, the content of this view is not updated when the STATISTICS_LEVEL initialization

parameter is set to BASIC. Base statistics for this view are reset at instance startup or when the value of the PGA_AGGREGATE_TARGET initialization parameter is dynamically modified.

Column	Datatype	Description
PGA_TARGET_FOR_ESTIMATE	NUMBER	Value of PGA_AGGREGATE_TARGET for this prediction (in bytes)
PGA_TARGET_FACTOR	NUMBER	PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2(3)	Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
LOW_OPTIMAL_SIZE	NUMBER	Lower bound for the optimal memory requirement of work areas included in this row (in bytes)
HIGH_OPTIMAL_SIZE	NUMBER	Upper bound for the optimal memory requirement of work areas included in this row (in bytes)
ESTD_OPTIMAL_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run optimal given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_ONEPASS_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run one-pass given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_MULTIPASSES_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run multi-pass given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_TOTAL_EXECUTIONS	NUMBER	Sum of ESTD_OPTIMAL_EXECUTIONS, ESTD_ONEPASS_EXECUTIONS, and ESTD_MULTIPASSES_EXECUTIONS
IGNORED_WORKAREAS_COUNT	NUMBER	Number of work areas with optimal memory requirement between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE ignored in the advice generation due to memory and CPU constraints

See Also: *Oracle Database Performance Tuning Guide* for information on tuning the PGA_AGGREGATE_TARGET initialization parameter using the PGA advice views

V\$PGASTAT

V\$PGASTAT provides PGA memory usage statistics as well as statistics about the automatic PGA memory manager when it is enabled (that is, when PGA_AGGREGATE_TARGET is set). Cumulative values in V\$PGASTAT are accumulated since instance startup.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the statistic (see Table 5-2)
VALUE	NUMBER	Statistic value
UNIT	VARCHAR2(12)	Unit for the value (microseconds, bytes, or percent)

Table 5–2 V\$PGASTAT Statistics

Statistic Name	Description
aggregate PGA target parameter	Current value of the <code>PGA_AGGREGATE_TARGET</code> initialization parameter. If this parameter is not set, then its value is 0 and automatic management of PGA memory is disabled.
aggregate PGA auto target	Amount of PGA memory the Oracle Database can use for work areas running in automatic mode. This amount is dynamically derived from the value of the <code>PGA_AGGREGATE_TARGET</code> initialization parameter and the current work area workload, and continuously adjusted by the Oracle Database. If this value is small compared to the value of <code>PGA_AGGREGATE_TARGET</code> , then a large amount of PGA memory is used by other components of the system (for example, PL/SQL or Java memory) and little is left for work areas. The DBA must ensure that enough PGA memory is left for work areas running in automatic mode.
global memory bound	Maximum size of a work area executed in automatic mode. This value is continuously adjusted by the Oracle Database to reflect the current state of the work area workload. The global memory bound generally decreases when the number of active work areas is increasing in the system. If the value of the global bound decreases below 1 MB, then the value of <code>PGA_AGGREGATE_TARGET</code> should be increased.
total PGA allocated	Current amount of PGA memory allocated by the instance. The Oracle Database attempts to keep this number below the value of the <code>PGA_AGGREGATE_TARGET</code> initialization parameter. However, it is possible for the PGA allocated to exceed that value by a small percentage and for a short period of time when the work area workload is increasing very rapidly or when <code>PGA_AGGREGATE_TARGET</code> is set to a small value.
total PGA used	Indicates how much PGA memory is currently consumed by work areas. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
total PGA used for auto workareas	Indicates how much PGA memory is currently consumed by work areas running under the automatic memory management mode. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
total PGA used for manual workareas	Indicates how much PGA memory is currently consumed by work areas running under the manual memory management mode. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
over allocation count	This statistic is cumulative since instance startup. Over allocating PGA memory can happen if the value of <code>PGA_AGGREGATE_TARGET</code> is too small. When this happens, the Oracle Database cannot honor the value of <code>PGA_AGGREGATE_TARGET</code> and extra PGA memory needs to be allocated. If over allocation occurs, then increase the value of <code>PGA_AGGREGATE_TARGET</code> using the information provided by the <code>V\$PGA_TARGET_ADVICE</code> view.
bytes processed	Number of bytes processed by memory intensive SQL operators, cumulated since instance startup.
extra bytes read/written	Number of bytes processed during extra passes of the input data, cumulated since instance startup. When a work area cannot run optimal, one or more of these extra passes is performed.
cache hit percentage	A metric computed by the Oracle Database to reflect the performance of the PGA memory component, cumulative since instance startup. A value of 100% means that all work areas executed by the system since instance startup have used an optimal amount of PGA memory. When a work area cannot run optimal, one or more extra passes is performed over the input data. This will reduce the cache hit percentage in proportion to the size of the input data and the number of extra passes performed.

V\$PQ_SESSTAT

This view lists session statistics for parallel queries. After you have run a query or DML operation, you can use the information derived from `V$PQ_SESSTAT` to view the number of slave processes used, and other information for the session and system.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic: <ul style="list-style-type: none"> ■ Queries Parallelized - Number of queries run in parallel ■ DDL Parallelized - Number of DDL operations run in parallel ■ DML Parallelized - Number of DML operations run in parallel ■ DFO Trees - Number of executed DFO trees ■ Server Threads - Total number of cluster databases used ■ Allocation Height - Requested number of servers per instance ■ Allocation Width - Requested number of instances ■ Local Msgs Sent - Number of local (intra-instance) messages sent ■ Distr Msgs Sent - Number of remote (inter-instance) messages sent ■ Local Msgs Recv'd - Number of local (intra-instance) messages received ■ Distr Msgs Recv'd - Number of remote (inter-instance) messages received
LAST_QUERY	NUMBER	Value of the statistic for the last operation
SESSION_TOTAL	NUMBER	Value of the statistic for the entire session to this point in time

V\$PQ_SLAVE

This view lists statistics for each of the active parallel execution servers on an instance.

Column	Datatype	Description
SLAVE_NAME	VARCHAR2(4)	Name of the parallel execution server
STATUS	VARCHAR2(4)	Current status of the parallel execution server (BUSY IDLE)
SESSIONS	NUMBER	Number of sessions that have used this parallel execution server
IDLE_TIME_CUR	NUMBER	Amount of time spent idle while processing statements in the current session
BUSY_TIME_CUR	NUMBER	Amount of time spent busy while processing statements in the current session
CPU_SECS_CUR	NUMBER	Amount of CPU time spent on the current session
MSGS_SENT_CUR	NUMBER	Number of messages sent while processing statements for the current session
MSGS_RCVD_CUR	NUMBER	Number of messages received while processing statements for the current session
IDLE_TIME_TOTAL	NUMBER	Total amount of time this query server has been idle
BUSY_TIME_TOTAL	NUMBER	Total amount of time this query server has been active
CPU_SECS_TOTAL	NUMBER	Total amount of CPU time this query server has used to process statements
MSGS_SENT_TOTAL	NUMBER	Total number of messages this query server has sent
MSGS_RCVD_TOTAL	NUMBER	Total number of messages this query server has received

V\$PQ_SYSSTAT

This view lists system statistics for parallel queries. After you have run a query or DML operation, you can use the information derived from V\$PQ_SYSSTAT to view the number of slave processes used, and other information for the system.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic: <ul style="list-style-type: none"> ▪ Servers Busy - Number of currently busy servers on this instance ▪ Servers Idle - Number of currently idle servers on this instance ▪ Servers Highwater - Number of active servers on this instance that have partaken in >= 1 operation so far ▪ Server Sessions - Total number of operations executed in all servers on this instance ▪ Servers Started - Total number of servers started on this instance ▪ Servers Shutdown - Total number of servers shutdown on this instance ▪ Servers Cleaned Up - Total number of servers on this instance cleaned up due to process death ▪ Queries Initiated - Total number of parallel queries initiated on this instance ▪ DDL Initiated - Total number of parallel DDL operations that were initiated ▪ DML Initiated - Total number of parallel DML operations that were initiated ▪ DFO Trees - Total number of DFO trees executed on this instance ▪ Local Msgs Sent - Total number of local (intra-instance) messages sent on this instance ▪ Distr Msgs Sent - Total number of remote (inter-instance) messages sent on this instance ▪ Local Msgs Recv'd - Total number of remote (inter-instance) messages received on this instance ▪ Distr Msgs Recv'd - Total number of remote (inter-instance) messages received on this instance
VALUE	NUMBER	Value of the statistic

V\$PQ_TQSTAT

This view contains statistics on parallel execution operations. The statistics are compiled after the query completes and only remain for the duration of the session. It displays the number of rows processed through each parallel execution server at each stage of the execution tree. This view can help determine skew problems in a query's execution.

Column	Datatype	Description
DFO_NUMBER	NUMBER	Data flow operator (DFO) tree number to differentiate queries
TQ_ID	NUMBER	Table queue ID within the query, which represents the connection between two DFO nodes in the query execution tree
SERVER_TYPE	VARCHAR2(10)	The role in table queue - producer/consumer/ranger
NUM_ROWS	NUMBER	The number of rows produced/consumed
BYTES	NUMBER	The number of bytes produced/consumed
OPEN_TIME	NUMBER	Time (seconds) the table queue remained open
AVG_LATENCY	NUMBER	Time (minutes) for a message to be dequeued after it enters the queue
WAITS	NUMBER	The number of waits encountered during dequeue
TIMEOUTS	NUMBER	The number of timeouts when waiting for a message
PROCESS	VARCHAR2(10)	Process ID
INSTANCE	NUMBER	Instance ID

V\$PROCESS

This view contains information about the currently active processes. While the LATCHWAIT column indicates what latch a process is waiting for, the LATCHSPIN column indicates what latch a process is spinning on. On multi-processor machines, Oracle processes will spin on a latch before waiting on it.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of process state object
PID	NUMBER	Oracle process identifier
SPID	VARCHAR2(12)	Operating system process identifier
USERNAME	VARCHAR2(15)	Operating system process username. Any two-task user coming across the network has "-T" appended to the username.
SERIAL#	NUMBER	Process serial number
TERMINAL	VARCHAR2(30)	Operating system terminal identifier
PROGRAM	VARCHAR2(48)	Program in progress
TRACEID	VARCHAR2(255)	Trace file identifier
BACKGROUND	VARCHAR2(1)	1 for a background process; NULL for a normal process
LATCHWAIT	VARCHAR2(8)	Address of latch the process is waiting for; NULL if none
LATCHSPIN	VARCHAR2(8)	Address of the latch the process is spinning on; NULL if none
PGA_USED_MEM	NUMBER	PGA memory currently used by the process
PGA_ALLOC_MEM	NUMBER	PGA memory currently allocated by the process (including free PGA memory not yet released to the operating system by the server process)
PGA_FREEABLE_MEM	NUMBER	Allocated PGA memory which can be freed
PGA_MAX_MEM	NUMBER	Maximum PGA memory ever allocated by the process

V\$PROPAGATION_RECEIVER

V\$PROPAGATION_RECEIVER displays information about buffer queue propagation schedules on the receiving (destination) side. The values are reset to zero when the database (or instance in a RAC environment) restarts, when propagation migrates to another instance, or when an unscheduled propagation is attempted.

Column	Datatype	Description
SRC_QUEUE_NAME	VARCHAR2(66)	Name of the source Queue
SRC_DBNAME	VARCHAR2(128)	Name of the source database
STARTUP_TIME	DATE	Startup time of this schedule. This time changes when the source/destination database gets restarted.
HIGH_WATER_MARK	NUMBER	High-watermark of the messages received
ACKNOWLEDGEMENT	NUMBER	Acknowledgement of the messages received by the receiver
ELAPSED_UNPICKLE_TIME	NUMBER	Elapsed unpickle time
ELAPSED_RULE_TIME	NUMBER	Elapsed rule time
ELAPSED_ENQUEUE_TIME	NUMBER	Elapsed enqueue time

V\$PROPAGATION_SENDER

V\$PROPAGATION_SENDER displays information about buffer queue propagation schedules on the sending (source) side. The values are reset to zero when the database (or instance in a RAC environment) restarts, when propagation migrates to another instance, or when an unscheduled propagation is attempted.

Column	Datatype	Description
QUEUE_ID	NUMBER	Queue identifier of the queue
QUEUE_SCHEMA	VARCHAR2(30)	Schema of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
DBLINK	VARCHAR2(128)	Name of the destination database link
HIGH_WATER_MARK	NUMBER	High-watermark of the messages sent
ACKNOWLEDGEMENT	NUMBER	Acknowledgement of the messages received by the receiver
SCHEDULE_STATUS	VARCHAR2(30)	Status of the propagation schedule
TOTAL_MSGS	NUMBER	Total messages propagated
TOTAL_BYTES	NUMBER	Total bytes propagated
ELAPSED_DEQUEUE_TIME	NUMBER	Elapsed dequeue time
ELAPSED_PICKLE_TIME	NUMBER	Elapsed pickle time (time taken to linearize an LCR into a stream of bytes that can be sent over the network)
ELAPSED_PROPAGATION_TIME	NUMBER	Elapsed propagation time

V\$PROXY_ARCHIVEDLOG

V\$PROXY_ARCHIVEDLOG contains descriptions of archived log backups which are taken with Proxy Copy. Each row represents a backup of one archived log.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record ID
STAMP	NUMBER	Proxy copy record stamp
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the copy resides
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. This value is informational only. It is not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.
TAG	VARCHAR2(32)	Proxy copy tag
STATUS	VARCHAR2(1)	Indicates the status of the copy: <ul style="list-style-type: none"> ■ A - Object is available ■ D - Object is deleted ■ X - Object has been "cross-checked" and found not to exist. A subsequent "delete expired" command will change the status to D. If, for some reason, the object really does still exist, then a subsequent "cross-check" command will change the status back to A.
DELETED	VARCHAR2(3)	Indicates whether the record has been deleted (YES) or not (NO)
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
FIRST_CHANGE#	NUMBER	First change number in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change number in the next log

Column	Datatype	Description
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log (in blocks)
BLOCK_SIZE	NUMBER	Redo log block size
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp

V\$PROXY_DATAFILE

This view contains descriptions of datafile and control file backups that are taken with Proxy Copy. Each row represents a backup of one database file.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record ID
STAMP	NUMBER	Proxy copy record stamp
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the copy resides
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. This value is informational only. It is not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command
TAG	VARCHAR2(32)	Proxy copy tag
STATUS	VARCHAR2(1)	Indicates the status of the copy: <ul style="list-style-type: none"> ■ A - Object is available ■ D - Object is deleted ■ X - Object has been "cross-checked" and found not to exist. A subsequent "delete expired" command will change the status to D. If, for some reason, the object really does still exist, then a subsequent "cross-check" command will change the status back to A.
DELETED	VARCHAR2(3)	Indicates whether the record has been deleted (YES) or not (NO)
FILE#	NUMBER	Absolute datafile number, or 0 if this is a control file backup
CREATION_CHANGE#	NUMBER	Datafile creation change number
CREATION_TIME	DATE	Datafile creation Timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change in any block of the file, if known
RECOVERY_FUZZY_CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY_TIME	DATE	Timestamp of the highest change written to the file by media recovery
INCREMENTAL_LEVEL	NUMBER	If this backup is part of an incremental backup strategy, then 0. Otherwise null.

Column	Datatype	Description
ONLINE_FUZZY	VARCHAR2 (3)	(YES NO) If set to YES, this copy was made after a crash or offline immediate (or is a copy of a copy which was taken improperly while the database was open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent.
BACKUP_FUZZY	VARCHAR2 (3)	(YES NO) If set to YES, this is a copy taken using the BEGIN BACKUP END BACKUP technique. Note that the BEGIN BACKUP END BACKUP technique is used internally when proxy copies of open files are created. Recovery will need to apply all redo up to the end backup marker to make this copy consistent.
BLOCKS	NUMBER	Size of the copy (in blocks). Also the size of the datafile when the copy was made.
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE_RANGE	NUMBER	If file# is 0 (that is, this is a control file backup), the RECID of the oldest offline range record in this control file copy. 0 for datafile copies.
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
CONTROLFILE_TYPE	VARCHAR2 (1)	B indicates normal copies S indicates standby copies
KEEP	VARCHAR2 (3)	(YES NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy
KEEP_UNTIL	DATE	If KEEP_UNTIL_TIME is specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2 (10)	Additional retention options for this backup set: <ul style="list-style-type: none"> ▪ LOGS - Logs need to recover this backup are kept ▪ NOLOGS - Logs needed to recover this backup will not be kept
RMAN_STATUS_RECID	NUMBER	Owning V\$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V\$RMAN_STATUS record stamp

V\$PWFILERS_USERS

This view lists users who have been granted SYSDBA and SYSOPER privileges as derived from the password file.

Column	Datatype	Description
USERNAME	VARCHAR2 (30)	The name of the user that is contained in the password file
SYSDBA	VARCHAR2 (5)	If TRUE, the user can connect with SYSDBA privileges
SYSOPER	VARCHAR2 (5)	If TRUE, the user can connect with SYSOPER privileges

V\$PX_PROCESS

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SERVER_NAME	VARCHAR2 (4)	The name of the cluster database (P000, P001, etc)
STATUS	VARCHAR2 (9)	The state of the cluster database (IN USE AVAILABLE)
PID	NUMBER	The process identifier
SPID	VARCHAR2 (12)	OS process ID
SID	NUMBER	The session ID of slave, if in use

Column	Datatype	Description
SERIAL#	NUMBER	The session serial number of slave, if in use

V\$PX_PROCESS_SYSSTAT

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	<p>Name of the statistic:</p> <ul style="list-style-type: none"> ▪ <code>Servers In Use</code> - Number of PX servers currently performing parallel operations ▪ <code>Servers Available</code> - Number of PX servers available to perform parallel operations ▪ <code>Servers Started</code> - Number of times the system has had to create a PX server process ▪ <code>Servers Shutdown</code> - Number of times a PX server process has been shut down. A PX server process will be shut down if it has not been used recently. The length of time it may remain "Available" is controlled by the <code>PARALLEL_SERVER_IDLE_TIME</code> initialization parameter. If this value is large, then consider increasing the parameter. This will improve performance by avoiding the latency of PX server process creation. ▪ <code>Servers HWM</code> - Maximum number of concurrent PX server processes If this number is equal to the <code>PARALLEL_MAX_SERVERS</code> initialization parameter, then consider increasing the parameter. This could allow you to increase your throughput, especially if your system is under-utilized and the <code>V\$SYSSTAT</code> statistic "Parallel operations downgraded to serial" is large. ▪ <code>Servers Cleaned Up</code> - Number of times PMON had to clean up a PX server. This should only happen during abnormal termination of a parallel operation. If this number is large, then you should determine the cause. ▪ <code>Server Sessions</code> - Total number of sessions created by all PX servers ▪ <code>Memory Chunks Allocated</code> - Number of large memory chunks allocated by PX servers ▪ <code>Memory Chunks Freed</code> - Number of large memory chunks freed ▪ <code>Memory Chunks Current</code> - Number of large memory chunks currently being used ▪ <code>Memory Chunks HWM</code> - Maximum number of concurrently allocated chunks ▪ <code>Buffers allocated</code> - Number of times a message buffer has been allocated ▪ <code>Buffers freed</code> - Number of times a message buffer has been freed ▪ <code>Buffers Current</code> - Number of message buffers currently being used ▪ <code>Buffers HWM</code> - Maximum number of concurrently allocated message buffers
VALUE	NUMBER	Value of the statistic

V\$PX_SESSION

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4 8)	Session address

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this cluster database process belongs
SERVER_SET	NUMBER	The logical set of servers to which this cluster database process belongs. A single server group will have at most two server sets.
SERVER#	NUMBER	The logical number of the cluster database process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions

V\$PX_SESSTAT

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4 8)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this cluster database process belongs
SERVER_SET	NUMBER	The logical set of servers that this cluster database process belongs to. A single server group will have at most two server sets.
SERVER#	NUMBER	The logical number of the cluster database process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions
STATISTIC#	NUMBER	Statistic number
VALUE	NUMBER	Statistic value

V\$QUEUE

V\$QUEUE contains information on the shared server message queues.

Column	Datatype	Description
PADDR	RAW(4 8)	Address of the process that owns the queue
TYPE	VARCHAR2(10)	Type of queue: <ul style="list-style-type: none"> ■ COMMON - Processed by servers ■ DISPATCHER
QUEUED	NUMBER	Number of items in the queue
WAIT	NUMBER	Total time that all items in this queue have waited (in hundredths of a second). Divide by TOTALQ for average wait per item.

Column	Datatype	Description
TOTALQ	NUMBER	Total number of items that have ever been in the queue

V\$QUEUEING_MTH

V\$QUEUEING_MTH displays all available queuing resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the queuing resource allocation method

V\$RECOVER_FILE

This view displays the status of files needing media recovery.

Column	Datatype	Description
FILE#	NUMBER	File identifier number
ONLINE	VARCHAR2(7)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in ONLINE_STATUS.
ONLINE_STATUS	VARCHAR2(7)	Online status (ONLINE, OFFLINE)
ERROR	VARCHAR2(18)	Why the file needs to be recovered: NULL if reason unknown, or OFFLINE NORMAL if recovery not needed
CHANGE#	NUMBER	SCN where recovery must start
TIME	DATE	Time of SCN when recovery must start

V\$RECOVERY_FILE_DEST

V\$RECOVERY_FILE_DEST displays information about the disk quota and current disk usage in the flash recovery area.

Column	Datatype	Description
NAME	VARCHAR2(513)	Location name. This is the value specified in the DB_RECOVERY_FILE_DEST initialization parameter.
SPACE_LIMIT	NUMBER	Maximum amount of disk space (in bytes) that the database can use for the flash recovery area. This is the value specified in the DB_RECOVERY_FILE_DEST_SIZE initialization parameter.
SPACE_USED	NUMBER	Amount of disk space (in bytes) used by flash recovery area files created in current and all previous flash recovery areas. Changing flash recovery areas does not reset SPACE_USED to 0.
SPACE_RECLAIMABLE	NUMBER	Total amount of disk space (in bytes) that can be created by deleting obsolete, redundant, and other low priority files from the flash recovery area
NUMBER_OF_FILES	NUMBER	Number of files in the flash recovery area

V\$RECOVERY_FILE_STATUS

V\$RECOVERY_FILE_STATUS contains one row for each datafile for each RECOVER statement. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V\$RECOVERY_FILE_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
FILENUM	NUMBER	Number of the file being recovered
FILENAME	VARCHAR2 (513)	Filename of the datafile being recovered
STATUS	VARCHAR2 (13)	Status of the recovery: <ul style="list-style-type: none"> ■ IN RECOVERY ■ CURRENT ■ NOT RECOVERED

See Also: *Oracle Database Backup and Recovery Basics*

V\$RECOVERY_LOG

This view lists information about archived logs that are needed to complete media recovery. This information is derived from the log history view, V\$LOG_HISTORY.

V\$RECOVERY_LOG contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V\$RECOVERY_LOG will be empty to all other Oracle users.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
TIME	DATE	Time of the first entry (lowest SCN) in the log
ARCHIVE_NAME	VARCHAR2 (513)	Name of the file when archived, using the naming convention specified by the LOG_ARCHIVE_FORMAT initialization parameter See Also: " LOG_ARCHIVE_FORMAT " on page 1-74

See Also: "[V\\$LOG_HISTORY](#)" on page 4-79 and *Oracle Database Backup and Recovery Advanced User's Guide*

V\$RECOVERY_PROGRESS

V\$RECOVERY_PROGRESS can be used to track database recovery operations to ensure that they are not stalled, and also to estimate the time required to complete the operation in progress.

V\$RECOVERY_PROGRESS is a subview of V\$SESSION_LONGOPS.

Column	Datatype	Description
TYPE	VARCHAR2 (64)	Type of recovery operation being performed
ITEM	VARCHAR2 (32)	Item being measured
SOFAR	NUMBER	Amount of work done so far
TOTAL	NUMBER	Total amount of work expected

See Also: *Oracle Database Backup and Recovery Advanced User's Guide*

V\$RECOVERY_STATUS

V\$RECOVERY_STATUS contains statistics of the current recovery process. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V\$RECOVERY_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
RECOVERY_CHECKPOINT	DATE	Point in time to which the recovery has occurred. If no logs have been applied, this is the point in time the recovery starts.
THREAD	NUMBER	Number of the redo thread currently being processed
SEQUENCE_NEEDED	NUMBER	Log sequence number of the log needed by the recovery process. The value is 0 if no log is needed.
SCN_NEEDED	VARCHAR2(16)	Low SCN of the log needed by recovery. The value is 0 if unknown or no log is needed.
TIME_NEEDED	DATE	Time when the log was created. The value is midnight on 1/1/88 if the time is unknown or if no log is needed.
PREVIOUS_LOG_NAME	VARCHAR2(513)	Filename of the log
PREVIOUS_LOG_STATUS	VARCHAR2(13)	Status of the previous log. Contains one of the following values: RELEASE; WRONG NAME; MISSING NAME; UNNEEDED NAME; NONE
REASON	VARCHAR2(13)	Reason recovery is returning control to the user (NEED LOG LOG REUSED THREAD DISABLED)

See Also: *Oracle Database Backup and Recovery Basics*

V\$REPLPROP

V\$REPLPROP displays information about the parallel propagation currently in progress at the replication site. Use this view to determine which transactions are currently being propagated, the number of calls propagated in each transaction, and the current activity of the parallel propagation slave processes or parallel propagation coordinator process.

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number. Used to identify uniquely a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends with, and another session begins with, the same session ID.
NAME	VARCHAR2(71)	Name of the propagation session: <ul style="list-style-type: none"> ■ Replication Parallel Prop Slaven - Slave process <i>n</i> is active, either waiting, pushing deferred transactions, purging metadata, or creating an error transaction ■ Replication Parallel Prop Coordinator - Coordinator process is active, and either waiting, sleeping, or scheduling slaves to perform operations <p>The coordinator process reads transactions from the deferred transaction queue and assigns them to the slave processes. Then, the slaves propagate the transactions to the destination site. When the slaves push transactions in a push session, the slaves remain active until the push session completes, even if there are no more transactions to push.</p>
DBLINK	VARCHAR2(128)	Database link on which this replication session is propagating

Column	Datatype	Description
STATE	VARCHAR2 (12)	State of the propagation session: <ul style="list-style-type: none"> ■ WAIT - Slave processes or the coordinator process is waiting for an event (that is, a message) ■ SLEEP - Coordinator process is sleeping for the duration of the <code>delay_seconds</code> setting. Note: <code>delay_seconds</code> is set with the <code>DBMS_DEFER_SYS.SCHEDULE_PUSH</code> procedure. ■ PUSH - Slave processes are pushing transactions from the deferred transaction queue to the remote site ■ PURGE - Slave processes are purging metadata related to successfully applied transactions from the remote site ■ CREATE ERROR - Slave processes are creating an error transaction. In this case, an error or a conflict occurred while the slaves were pushing deferred transactions to the remote site. ■ SCHEDULE TXN - Coordinator process is determining the order that transactions are applied and assigning slave processes to execute the transactions
XID	VARCHAR2 (22)	If the session is a slave session, then indicates the transaction id of the transaction that the slave is currently propagating.
SEQUENCE	NUMBER	If the process is a slave process, then indicates the sequence number of the calls propagated in the current operation, if relevant. Each transaction must process one or more calls, and the value of <code>SEQUENCE</code> starts at zero and increases as each call is processed. The <code>SEQUENCE</code> value shows the call that is currently being processed in each transaction. This value increases until the slave has processed all of the calls in a transaction.

Note: This view only contains data when deferred transactions are being pushed using parallel propagation at the current site. The `parallelism` parameter must be set to 1 or higher in the `DBMS_DEFER_SYS.PUSH` function for a push to use parallel propagation. Otherwise, the push uses serial propagation, and no data appears in this view during the push.

V\$REPLQUEUE

V\$REPLQUEUE displays statistics about the replication deferred transactions queue. All values are stored since the start of the current database instance.

Column	Datatype	Description
TXNS_ENQUEUED	NUMBER	Number of transactions enqueued in the deferred transactions queue
CALLS_ENQUEUED	NUMBER	Number of calls enqueued into the deferred transactions queue
TXNS_PURGED	NUMBER	Number of transactions purged from the deferred transactions queue
LAST_ENQUEUE_TIME	DATE	Date when the last transaction was enqueued into the deferred transaction queue (null if no transactions have been enqueued into the deferred transaction queue since the instance started)
LAST_PURGE_TIME	DATE	Date when the last transaction was purged from the deferred transaction queue (null if no transactions have been purged from the deferred transaction queue since the instance started)

V\$REQDIST

This view lists statistics for the histogram of shared server dispatcher request times, divided into 12 buckets, or ranges of time. The time ranges grow exponentially as a function of the bucket number.

Column	Datatype	Description
BUCKET	NUMBER	Bucket number: 0 - 11; the maximum time for each bucket is $(4 * 2^N) / 100$ seconds
COUNT	NUMBER	Count of requests whose total time to complete (excluding wait time) falls in this range

V\$RESERVED_WORDS

This view gives a list of all the keywords that are used by the PL/SQL compiler. This view helps developers to determine whether a word is already being used as a keyword in the language.

Column	Datatype	Description
KEYWORD	VARCHAR2(30)	Name of the keyword
LENGTH	NUMBER	Length of the keyword

V\$RESOURCE

This view contains resource name and address information.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of the resource object
TYPE	VARCHAR2(2)	Resource type; the resource types are listed in Table 4-1 on page 4-77
ID1	NUMBER	Resource identifier #1
ID2	NUMBER	Resource identifier #2

V\$RESOURCE_LIMIT

This view displays information about global resource use for some of the system resources. Use this view to monitor the consumption of resources so that you can take corrective action, if necessary. Many of the resources correspond to initialization parameters listed in [Table 5-3](#).

Some resources, those used by DLM for example, have an initial allocation (soft limit), and the hard limit, which is theoretically infinite (although in practice it is limited by SGA size). During SGA reservation/initialization, a place is reserved in SGA for the `INITIAL_ALLOCATION` of resources, but if this allocation is exceeded, additional resources are allocated up to the value indicated by `LIMIT_VALUE`. The `CURRENT_UTILIZATION` column indicates whether the initial allocation has been exceeded. When the initial allocation value is exceeded, the additional required resources are allocated from the shared pool, where they must compete for space with other resources.

A good choice for the value of `INITIAL_ALLOCATION` will avoid the contention for space. For most resources, the value for `INITIAL_ALLOCATION` is the same as the `LIMIT_VALUE`. Exceeding `LIMIT_VALUE` results in an error.

Column	Datatype	Description
RESOURCE_NAME	VARCHAR2(30)	Name of the resource (see Table 5-3)
CURRENT_UTILIZATION	NUMBER	Number of (resources, locks, or processes) currently being used
MAX_UTILIZATION	NUMBER	Maximum consumption of this resource since the last instance start-up

Column	Datatype	Description
INITIAL_ALLOCATION	VARCHAR2(10)	Initial allocation. This will be equal to the value specified for the resource in the initialization parameter file (UNLIMITED for infinite allocation).
LIMIT_VALUE	VARCHAR2(10)	Unlimited for resources and locks. This can be greater than the initial allocation value (UNLIMITED for infinite limit).

Table 5-3 Values for the RESOURCE_NAME Column

Resource Name	Corresponds to
DML_LOCKS	See "DML_LOCKS" on page 1-50
ENQUEUE_LOCKS	This value is computed by the Oracle Database. See V\$ENQUEUE_LOCK on page 4-55 to obtain more information about the enqueue locks.
ENQUEUE_RESOURCES	See "ENQUEUE_RESOURCES" on page 1-51
GES_LOCKS	Global Enqueue Service locks
GES_PROCS	Global Enqueue Service processes
GES_RESS	Global Enqueue Service resources
MAX_SHARED_SERVERS	See "MAX_SHARED_SERVERS" on page 1-85
PARALLEL_MAX_SERVERS	See "PARALLEL_MAX_SERVERS" on page 1-104
PROCESSES	See "PROCESSES" on page 1-112
SESSIONS	See "SESSIONS" on page 1-124
SORT_SEGMENT_LOCKS	This value is computed by the Oracle Database
TEMPORARY_LOCKS	This value is computed by the Oracle Database
TRANSACTIONS	See "TRANSACTIONS" on page 1-141

V\$RMAN_CONFIGURATION

This view lists information about RMAN persistent configuration settings.

Column	Datatype	Description
CONF#	NUMBER	A unique key identifying this record within the database that owns it
NAME	VARCHAR2(65)	Name for this particular configuration. Example: RETENTION POLICY
VALUE	VARCHAR2(1025)	Value for this configuration row. Example: RETENTION POLICY TO RECOVERY WINDOW OF 10 DAYS

V\$RMAN_OUTPUT

V\$RMAN_OUTPUT displays messages reported by RMAN. This is an in-memory view and is not recorded in the controlfile. The view can hold 32768 rows.

Column	Datatype	Description
SID	NUMBER	Session ID of the session which is running this RMAN operation
RECID	NUMBER	Record ID of the corresponding V\$RMAN_STATUS row
STAMP	NUMBER	Timestamp of the corresponding V\$RMAN_STATUS row
SESSION_RECID	NUMBER	Record ID of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
SESSION_STAMP	NUMBER	Timestamp of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
OUTPUT	VARCHAR2(129)	Output text reported by RMAN

V\$RMAN_STATUS

V\$RMAN_STATUS displays the finished and on-going RMAN jobs. For on-going jobs, this view displays progress and status. The jobs which are in progress are stored only in memory while the finished jobs are stored in the controlfile.

Column	Datatype	Description
SID	NUMBER	Session ID of the session which is running this RMAN operation
RECID	NUMBER	Record ID of the row in the controlfile
STAMP	NUMBER	Timestamp of the row (RECID + STAMP is unique)
PARENT_RECID	VARCHAR2(40)	Record ID of the parent row of this row (corresponding V\$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
PARENT_STAMP	VARCHAR2(40)	Timestamp of the parent row of this row (corresponding V\$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
SESSION_RECID	NUMBER	Record ID of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
SESSION_STAMP	NUMBER	Timestamp of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
ROW_LEVEL	NUMBER	Level of the row. The session has level 0.
ROW_TYPE	VARCHAR2(19)	Type of the row: <ul style="list-style-type: none"> ■ SESSION ■ COMMAND ■ RECURSIVE OPERATION
COMMAND_ID	VARCHAR2(33)	Command ID set by the RMAN SET COMMAND ID command. If not set, then RMAN will create a unique number.
OPERATION	VARCHAR2(33)	Name of the command in the execution explained by this row
STATUS	VARCHAR2(23)	Status of the operation: <ul style="list-style-type: none"> ■ RUNNING ■ RUNNING WITH WARNINGS ■ RUNNING WITH ERRORS ■ COMPLETED ■ COMPLETED WITH WARNINGS ■ COMPLETED WITH ERRORS ■ FAILED
MBYTES_PROCESSED	NUMBER	Percentage of the job completed; null if not applicable for the operation
START_TIME	DATE	Start time of the job
END_TIME	DATE	End time of the job

V\$ROLLNAME

This view lists the names of all online rollback segments. It can only be accessed when the database is open.

Column	Datatype	Description
USN	NUMBER	Rollback (undo) segment number
NAME	VARCHAR2(30)	Rollback segment name

V\$ROLLSTAT

This view contains rollback segment statistics.

Column	Datatype	Description
USN	NUMBER	Rollback segment number
LATCH	NUMBER	Latch for the rollback segment
EXTENTS	NUMBER	Number of extents in the rollback segment
RSSIZE	NUMBER	Size (in bytes) of the rollback segment. This value differs by the number of bytes in one database block from the value of the BYTES column of the ALL/DBA/USER_SEGMENTS views. See Also: <i>Oracle Database Administrator's Guide</i> .
WRITES	NUMBER	Number of bytes written to the rollback segment
XACTS	NUMBER	Number of active transactions
GETS	NUMBER	Number of header gets
WAITS	NUMBER	Number of header waits
OPTSIZE	NUMBER	Optimal size of the rollback segment
HWMSIZE	NUMBER	High-watermark of rollback segment size
SHRINKS	NUMBER	Number of times the size of a rollback segment decreases
WRAPS	NUMBER	Number of times rollback segment is wrapped
EXTENDS	NUMBER	Number of times rollback segment size is extended
AVESHRINK	NUMBER	Average shrink size
AVERAGEACTIVE	NUMBER	Current size of active extents, averaged over time.
STATUS	VARCHAR2(15)	Rollback segment status: <ul style="list-style-type: none"> ▪ ONLINE ▪ PENDING OFFLINE ▪ OFFLINE ▪ FULL
CUREXT	NUMBER	Current extent
CURBLK	NUMBER	Current block

V\$ROWCACHE

This view displays statistics for data dictionary activity. Each row contains statistics for one data dictionary cache.

Column	Datatype	Description
CACHE#	NUMBER	Row cache ID number
TYPE	VARCHAR2(11)	Parent or subordinate row cache type
SUBORDINATE#	NUMBER	Subordinate set number
PARAMETER	VARCHAR2(32)	Name of the initialization parameter that determines the number of entries in the data dictionary cache
COUNT	NUMBER	Total number of entries in the cache
USAGE	NUMBER	Number of cache entries that contain valid data
FIXED	NUMBER	Number of fixed entries in the cache
GETS	NUMBER	Total number of requests for information on the data object
GETMISSES	NUMBER	Number of data requests resulting in cache misses
SCANS	NUMBER	Number of scan requests
SCANMISSES	NUMBER	Number of times a scan failed to find the data in the cache
SCANCOMPLETES	NUMBER	For a list of subordinate entries, the number of times the list was scanned completely

Column	Datatype	Description
MODIFICATIONS	NUMBER	Number of inserts, updates, and deletions
FLUSHES	NUMBER	Number of times flushed to disk
DLM_REQUESTS	NUMBER	Number of DLM requests
DLM_CONFLICTS	NUMBER	Number of DLM conflicts
DLM_RELEASES	NUMBER	Number of DLM releases

V\$ROWCACHE_PARENT

This view displays information for parent objects in the data dictionary. There is one row per lock owner, and one waiter for each object. This row shows the mode held or requested. For objects with no owners or waiters, a single row is displayed.

Column	Datatype	Description
INDEX	NUMBER	Index of the row
HASH	NUMBER	Hash value
ADDRESS	RAW(4 8)	Address of the parent object
CACHE#	NUMBER	Parent cache ID
CACHE_NAME	VARCHAR2(64)	Parent cache name
EXISTENT	VARCHAR2(1)	Indicates whether the object is an existing object
LOCK_MODE	NUMBER	Mode the lock is held in
LOCK_REQUEST	NUMBER	Mode the lock is requested in
TXN	RAW(4)	Transaction currently locking the object
SADDR	RAW(4 8)	Address of the session
Note: The following are only relevant for Real Application Clusters.		
INST_LOCK_REQUEST	NUMBER	Mode in which instance lock is being requested
INST_LOCK_RELEASE	NUMBER	Whether the instance lock needs to be released
INST_LOCK_TYPE	VARCHAR2(2)	Type of instance lock
INST_LOCK_ID1	RAW(4)	ID associated with the instance lock
INST_LOCK_ID2	RAW(4)	ID associated with the instance lock
KEY	RAW(100)	Contents of the key

V\$ROWCACHE_SUBORDINATE

This view displays information for subordinate objects in the data dictionary.

Column	Datatype	Description
INDEX	NUMBER	The index
HASH	NUMBER	The hash value
ADDRESS	RAW(4 8)	Address of the subordinate object
CACHE#	NUMBER	The parent cache ID
SUBCACHE#	NUMBER	The subcache ID
SUBCACHE_NAME	VARCHAR2(64)	The subcache name
EXISTENT	VARCHAR2(1)	Whether the object is an existing object
PARENT	RAW(4 8)	Address of the parent object
KEY	RAW(100)	The contents of the key

V\$RSRC_CONSUMER_GROUP

This view displays data related to currently active resource consumer groups.

Column	Datatype	Description
NAME	VARCHAR2 (32)	Name of the consumer group
ACTIVE_SESSIONS	NUMBER	Number of currently active sessions in the consumer group
EXECUTION_WAITERS	NUMBER	Number of currently active sessions waiting for an execution time slice in which they will be able to use CPU.
REQUESTS	NUMBER	Cumulative number of requests that were executed in the consumer group
CPU_WAIT_TIME	NUMBER	Cumulative amount of time that sessions waited for CPU
CPU_WAITS	NUMBER	Cumulative number of times all sessions in the consumer group had to wait for CPU
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in the consumer group
YIELDS	NUMBER	Cumulative number of times sessions in the consumer group had to yield the CPU
QUEUE_LENGTH	NUMBER	Number of sessions waiting in the queue
CURRENT_UNDO_CONSUMPTION	NUMBER	Current amount (in kilobytes) of undo consumed by the consumer group

See Also:

- *Oracle Database Administrator's Guide* for information on resource groups
- *PL/SQL Packages and Types Reference* for information on creating resource groups with the DBMS_RESOURCE_MANAGER package

V\$RSRC_CONSUMER_GROUP_CPU_MTH

This view lists all resource allocation methods defined for resource consumer groups.

See Also:

- *Oracle Database Administrator's Guide* for information on resource allocation methods
- *PL/SQL Packages and Types Reference* on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package
- "[V\\$RSRC_PLAN_CPU_MTH](#)" on page 5-30 for a listing of all resource allocation methods defined for resource plans

Column	Datatype	Description
NAME	VARCHAR2 (40)	Name of the CPU resource allocation method

V\$RSRC_PLAN

This view displays the names of all currently active resource plans.

See Also:

- ["DBA_RSRC_PLANS"](#) on page 3-114 for a listing of all plans in the database
- *Oracle Database Administrator's Guide* for information on resource plans
- *PL/SQL Packages and Types Reference* on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package

Column	Datatype	Description
NAME	VARCHAR2 (32)	Name of the resource plan
IS_TOP_PLAN	VARCHAR2 (5)	Indicates whether the resource plan is the current top plan (TRUE) or the resource plan is a subplan of the current top plan (FALSE)

V\$RSRC_PLAN_CPU_MTH

This view lists all available CPU resource allocation methods defined for resource plans.

See Also:

- ["V\\$RSRC_CONSUMER_GROUP_CPU_MTH"](#) on page 5-29 for a listing of resource allocation methods defined for consumer groups
- *Oracle Database Administrator's Guide* for information on resource plans
- *PL/SQL Packages and Types Reference* on defining resource allocation methods for consumer plans with the DBMS_RESOURCE_MANAGER package

Column	Datatype	Description
NAME	VARCHAR2 (32)	Name of the resource allocation method

V\$RULE

V\$RULE displays rule statistics. This view has a row for every rule loaded into shared memory.

Column	Datatype	Description
RULE_SET_OBJECT_ID	NUMBER	Rule set object ID
EVALUATION_CONTEXT_OBJECT_ID	NUMBER	Evaluation context object ID
RULE_OWNER	VARCHAR2 (64)	Owner of the rule
RULE_NAME	VARCHAR2 (64)	Name of the rule
RULE_CONDITION	VARCHAR2 (64)	Rule condition
TRUE_HITS	NUMBER	Number of times the rule evaluated to TRUE
MAYBE_HITS	NUMBER	Number of times the rule evaluated to MAYBE
SQL_EVALUATIONS	NUMBER	Number of evaluations of the rule that were performed by issuing SQL

V\$RULE_SET

V\$RULE_SET displays rule set statistics. This view has a row for every rule set loaded into shared memory.

Column	Datatype	Description
OWNER	VARCHAR2(64)	Owner of the rule set
NAME	VARCHAR2(1000)	Name of the rule set
CPU_TIME	NUMBER	Total CPU time (in hundredths of a second) spent in evaluation of the rule set
ELAPSED_TIME	NUMBER	Total elapsed time (in hundredths of a second) spent in evaluation of the rule set
FIRST_LOAD_TIME	DATE	First time the current cached copy of the rule set was loaded
LAST_LOAD_TIME	DATE	Last time the current cached copy of the rule set was loaded
LAST_LOADING_TIME	NUMBER	Total elapsed time (in hundredths of a second) spent to load the rule set the last time it was loaded
SHARABLE_MEM	NUMBER	Shared memory (in bytes) used by the rule set
RELOADS	NUMBER	Number of times the rule set object was reloaded in shared memory
INVALIDATIONS	NUMBER	Number of times the rule set object was invalidated
EVALUATIONS	NUMBER	Number of evaluations on the rule set
FIRST_HIT_EVALUATIONS	NUMBER	Number of evaluations on the rule set, with <code>stop_on_first_hit</code> set to TRUE
SIMPLE_RULES_ONLY_EVALUATIONS	NUMBER	Number of evaluations on the rule set, with <code>simple_rules_only</code> set to TRUE
SQL_FREE_EVALUATIONS	NUMBER	Number of evaluations on the rule set which did not internally issue SQL to evaluate rules
SQL_EXECUTIONS	NUMBER	Total number of SQL statements executed during evaluation of the rule set
CONDITIONS_PROCESSED	NUMBER	Total number of fast (indexed) conditions processed during evaluation of the rule set
TRUE_RULES	NUMBER	Total number of TRUE rules returned during evaluation of the rule set
MAYBE_RULES	NUMBER	Total number of MAYBE rules returned during evaluation of the rule set
VARIABLE_VALUE_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined functions to retrieve variable values (specified by the <code>variable_value_function</code> field in <code>RE\$VARIABLE_TYPE</code>) made during evaluation of the rule set
VARIABLE_METHOD_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined functions to retrieve variable method values (specified by the <code>variable_method_function</code> field in <code>RE\$VARIABLE_TYPE</code>) made during evaluation of the rule set
EVALUATION_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined evaluation functions (specified as the <code>evaluation_function</code> argument to the <code>DBMS_RULE_ADM.CREATE_EVALUATION_CONTEXT</code> procedure) made during evaluation of the rule set

V\$RULE_SET_AGGREGATE_STATS

V\$RULE_SET_AGGREGATE_STATS displays statistics aggregated over all evaluations on all rule sets. This view has a row for each type of statistic.

Column	Datatype	Description
NAME	VARCHAR2(80)	Name of the statistic: <ul style="list-style-type: none"> ■ rule set evaluations (all) - Total number of evaluations on all rule sets ■ rule set evaluations (first_hit) - Total number of evaluations on rule sets with stop_on_first_hit set to TRUE ■ rule set evaluations (simple_rules_only) - Total number of evaluations on rule sets with simple_rules_only set to TRUE ■ rule set evaluations (SQL free) - Total number of evaluations on rule sets which did not internally issue SQL to evaluate rules ■ rule set evaluation time (CPU) - Total CPU time (in hundredths of a second) spent in evaluations on rule sets ■ rule set evaluation time (elapsed) - Total elapsed time (in hundredths of a second) spent in evaluations on rule sets ■ rule set SQL executions - Total number of SQL statements executed during evaluations on rule sets ■ rule set conditions processed - Total number of fast (indexed) conditions processed during evaluations on rule sets ■ rule set true rules - Total number of TRUE rules returned during evaluations on rule sets ■ rule set maybe rules - Total number of MAYBE rules returned during evaluations on rule sets ■ rule set user function calls (variable value function) - Total number of calls made to user-defined functions to retrieve variable values (specified by the variable_value_ function field in RE\$VARIABLE_TYPE) made during evaluations on rule sets ■ rule set user function calls (variable method function) - Total number of calls made to user-defined functions to retrieve variable method values (specified by the variable_ method_function field in RE\$VARIABLE_TYPE) made during evaluations on rule sets ■ rule set user function calls (evaluation function) - Total number of calls made to user-defined evaluation functions (specified as the evaluation_function argument to the DBMS_RULE_ADM.CREATE_EVALUATION_CONTEXT procedure) made during evaluations on rule sets
VALUE	NUMBER	Statistic value

V\$SEGMENT_STATISTICS

V\$SEGMENT_STATISTICS displays information about segment-level statistics.

Column	Datatype	Description
OWNER	VARCHAR2(30)	Owner of the object
OBJECT_NAME	VARCHAR2(30)	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)	Name of the subobject
TABLESPACE_NAME	VARCHAR2(30)	Name of the table space to which the object belongs
TS#	NUMBER	Tablespace number
OBJ#	NUMBER	Dictionary object number of the object
DATAOBJ#	NUMBER	Data object number of the object
OBJECT_TYPE	VARCHAR2(18)	Type of the object
STATISTIC_NAME	VARCHAR2(64)	Name of the statistic
STATISTIC#	NUMBER	Statistic number

Column	Datatype	Description
VALUE	NUMBER	Statistic value

V\$SEGSTAT

V\$SEGSTAT displays information about segment-level statistics.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
OBJ#	NUMBER	Dictionary object number
DATAOBJ#	NUMBER	Data object number
STATISTIC_NAME	VARCHAR2 (64)	Name of the statistic
STATISTIC#	NUMBER	Statistic number
VALUE	NUMBER	Statistic value

V\$SEGSTAT_NAME

V\$SEGSTAT_NAME displays information about segment-level statistics properties.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2 (64)	Name of the statistic
SAMPLED	VARCHAR2 (3)	Indicates whether the statistic was collected by sampling (YES) or not (NO)

V\$SERV_MOD_ACT_STATS

V\$SERV_MOD_ACT_STATS displays the same set of performance statistics as V\$SERVICE_STATS except for a specific combination of service/module/action names.

When aggregation is enabled for the service name, module, and action name, then this view provides the timing and work done for calls issued for the business transaction.

Column	Datatype	Description
AGGREGATION_TYPE	VARCHAR2 (21)	Aggregation statistic type: <ul style="list-style-type: none"> ■ SERVICE_MODULE - Action value is null and the entry is an aggregate for all actions within a given module ■ SERVICE_MODULE_ACTION - Action value is null only for an empty action, and the aggregation is on the level of service/module/action
SERVICE_NAME	VARCHAR2 (64)	Service name from V\$SERVICES
MODULE	VARCHAR2 (49)	Module name from DBA_ENABLED_AGGREGATIONS
ACTION	VARCHAR2 (33)	Action name from DBA_ENABLED_AGGREGATIONS
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2 (64)	Derived statistic name from V\$STATNAME and V\$SESS_TIME_MODEL%
VALUE	NUMBER	Cumulative value

V\$SERVICE_EVENT

V\$SERVICE_EVENT displays aggregated wait counts and wait times for each wait statistic.

V\$SERVICE_STATS

Column	Datatype	Description
SERVICE_NAME	VARCHAR2 (64)	Service name from V\$SERVICES
SERVICE_NAME_HASH	NUMBER	Service name hash from V\$SERVICES
EVENT	VARCHAR2 (64)	Name of the wait event; derived statistic name from V\$EVENT_NAME
EVENT_ID	NUMBER	Identifier of the event
TOTAL_WAITS	NUMBER	Total amount of time waited for the event by this service (in hundredths of a second)
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event by this service
TIME_WAITED	NUMBER	Time waited for the event (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event by this service (in hundredths of a second)
MAX_WAIT	NUMBER	Maximum time (in hundredths of a second) waited for the event by this service
TIME_WAITED_MICRO	NUMBER	Total time waited for the event (in microseconds)

V\$SERVICE_STATS

V\$SERVICE_STATS displays a minimal set of performance statistics. These call rate statistics are used for making run-time routing decisions, for tracking service levels, and for per-instance diagnostics per call rate. The elapsed timing for each call provides a relative value across instances for how well a node is processing SQL calls issued under a service name.

When aggregation is enabled for the Service Name, then this view provides the timing and work done for calls issued for the whole service.

Column	Datatype	Description
SERVICE_NAME_HASH	NUMBER	Service name hash from V\$SERVICES
SERVICE_NAME	VARCHAR2 (64)	Service name from V\$SERVICES
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2 (64)	Derived statistic name from V\$STATNAME and V\$SESS_TIME_MODEL%
VALUE	NUMBER	Cumulative value

V\$SERVICE_WAIT_CLASS

V\$SERVICE_WAIT_CLASS displays aggregated wait counts and wait times for each wait statistic. An aggregation of these wait classes is used when thresholds are imported.

Column	Datatype	Description
SERVICE_NAME	VARCHAR2 (64)	Service name from V\$SERVICES
SERVICE_NAME_HASH	NUMBER	Service name hash from V\$SERVICES
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2 (64)	Name of the wait class
TOTAL_WAITS	NUMBER	Amount of time spent in the class by this session
TIME_WAITED	NUMBER	Number of times waits of the class occurred for this client

V\$SERVICEMETRIC

V\$SERVICEMETRIC displays values of metrics for the most recent 60-second interval. A history of the last one hour will be kept in the system.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
SERVICE_NAME_HASH	NUMBER	Service name hash
SERVICE_NAME	VARCHAR2 (64)	Service name
CTMHASH	NUMBER	CTM hash
ELAPSEDPERCALL	NUMBER	Elapsed per call
CPUPERCALL	NUMBER	CPU per call

V\$SERVICEMETRIC_HISTORY

V\$SERVICEMETRIC_HISTORY displays values of metrics for all intervals in the last one hour.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
SERVICE_NAME_HASH	NUMBER	Service name hash
SERVICE_NAME	VARCHAR2 (64)	Service name
CTMHASH	NUMBER	CTM hash
ELAPSEDPERCALL	NUMBER	Elapsed per call
CPUPERCALL	NUMBER	CPU per call

V\$SERVICES

V\$SERVICES displays the services in the database.

Column	Datatype	Description
SERVICE_ID	NUMBER	Service ID
NAME	VARCHAR2 (64)	Name of the service
NAME_HASH	NUMBER	Hash of the service name
NETWORK_NAME	VARCHAR2 (512)	Network name
CREATION_DATE	DATE	Creation date
CREATION_DATE_HASH	NUMBER	Creation date hash

V\$SES_OPTIMIZER_ENV

V\$SES_OPTIMIZER_ENV displays the contents of the optimizer environment used by each session. When a new session is first created, it automatically inherits its optimizer

environment from the optimizer environment defined at the instance level by V\$SYS_OPTIMIZER_ENV. The value of certain parameters can be dynamically modified by issuing an ALTER SESSION statement.

Column	Datatype	Description
SID	NUMBER	Session identifier. This column can be used to join with V\$SESSION on the SID column.
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2(40)	Name of the parameter
ISDEFAULT	VARCHAR2(3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2(25)	Value of the parameter for the session

See Also: "[V\\$SYS_OPTIMIZER_ENV](#)" on page 5-78

V\$SESS_IO

This view lists I/O statistics for each user session.

Column	Datatype	Description
SID	NUMBER	Session identifier
BLOCK_GETS	NUMBER	Block gets for this session
CONSISTENT_GETS	NUMBER	Consistent gets for this session
PHYSICAL_READS	NUMBER	Physical reads for this session
BLOCK_CHANGES	NUMBER	Block changes for this session
CONSISTENT_CHANGES	NUMBER	Consistent changes for this session

V\$SESS_TIME_MODEL

V\$SESS_TIME_MODEL displays the session-accumulated time for various operations. The time reported is the total elapsed or CPU time (in microseconds). Any timed operation will buffer at most 5 seconds of time data. Specifically, this means that if a timed operation (such as SQL execution) takes a long period of time to perform, the data published to this view is at most missing 5 seconds of the time accumulated for the operation.

The time values are 8-byte integers and can therefore hold approximately 580,000 years of time before wrapping. Background process time is not included in a statistic value unless the statistic is specifically for background processes.

Column	Datatype	Description
SID	NUMBER	Session ID (same as in V\$SESSION)
STAT_ID	NUMBER	Statistic identifier for the time statistic
STAT_NAME	VARCHAR2(64)	Name of the statistic (see Table 5-4)
VALUE	NUMBER	Amount of time (in microseconds) that the session has spent in this operation

Table 5–4 V\$SESS_TIME_MODEL and V\$SYS_TIME_MODEL Statistics

Statistic Name	Description
DB Time	Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.
DB CPU	Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.
background cpu time	Amount of CPU time (in microseconds) consumed by database background processes.
sequence load elapsed time	Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.
parse time elapsed	Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.
hard parse elapsed time	Amount of elapsed time spent hard parsing SQL statements.
sql execute elapsed time	Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.
connection management call elapsed time	Amount of elapsed time spent performing session connect and disconnect calls.
failed parse elapsed time	Amount of time spent performing SQL parses which ultimately fail with some parse error.
hard parse (sharing criteria) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.
hard parse (bind mismatch) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.
PL/SQL execution elapsed time	Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.
PL/SQL compilation elapsed time	Amount of elapsed time spent running the PL/SQL compiler.
inbound PL/SQL rpc elapsed time	Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to "PL/SQL execution elapsed time".
Java execution elapsed time	Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.

V\$SESSION

This view lists session information for each current session.

Column	Datatype	Description
SADDR	RAW(4 8)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number. Used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends and another session begins with the same session ID.
AUDSID	NUMBER	Auditing session ID
PADDR	RAW(4 8)	Address of the process that owns the session
USER#	NUMBER	Oracle user identifier
USERNAME	VARCHAR2(30)	Oracle username
COMMAND	NUMBER	Command in progress (last statement parsed); for a list of values, see Table 5–5 . These values also appear in the AUDIT_ACTIONS table.

Column	Datatype	Description
OWNERID	NUMBER	The column contents are invalid if the value is 2147483644. Otherwise, this column contains the identifier of the user who owns the migratable session. For operations using Parallel Slaves, interpret this value as a 4-byte value. The low-order 2 bytes of which represent the session number, and the high-order bytes the instance ID of the query coordinator.
TADDR	VARCHAR2(8)	Address of transaction state object
LOCKWAIT	VARCHAR2(8)	Address of lock waiting for; null if none
STATUS	VARCHAR2(8)	Status of the session: <ul style="list-style-type: none"> ■ ACTIVE - Session currently executing SQL ■ INACTIVE ■ KILLED - Session marked to be killed ■ CACHED - Session temporarily cached for use by Oracle*XA ■ SNIPED - Session inactive, waiting on the client
SERVER	VARCHAR2(9)	Server type (DEDICATED SHARED PSEUDO NONE)
SCHEMA#	NUMBER	Schema user identifier
SCHEMANAME	VARCHAR2(30)	Schema user name
OSUSER	VARCHAR2(30)	Operating system client user name
PROCESS	VARCHAR2(12)	Operating system client process ID
MACHINE	VARCHAR2(64)	Operating system machine name
TERMINAL	VARCHAR2(30)	Operating system terminal name
PROGRAM	VARCHAR2(48)	Operating system program name
TYPE	VARCHAR2(10)	Session type
SQL_ADDRESS	RAW(4 8)	Used with SQL_HASH_VALUE to identify the SQL statement that is currently being executed
SQL_HASH_VALUE	NUMBER	Used with SQL_ADDRESS to identify the SQL statement that is currently being executed
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement that is currently being executed
SQL_CHILD_NUMBER	NUMBER	Child number of the SQL statement that is currently being executed
PREV_SQL_ADDR	RAW(4 8)	Used with PREV_HASH_VALUE to identify the last SQL statement executed
PREV_HASH_VALUE	NUMBER	Used with SQL_HASH_VALUE to identify the last SQL statement executed
PREV_SQL_ID	VARCHAR2(13)	SQL identifier of the last SQL statement executed
PREV_CHILD_NUMBER	NUMBER	Child number of the last SQL statement executed
MODULE	VARCHAR2(48)	Name of the currently executing module as set by calling the DBMS_APPLICATION_INFO.SET_MODULE procedure
MODULE_HASH	NUMBER	Hash value of the above MODULE
ACTION	VARCHAR2(32)	Name of the currently executing action as set by calling the DBMS_APPLICATION_INFO.SET_ACTION procedure
ACTION_HASH	NUMBER	Hash value of the above action name
CLIENT_INFO	VARCHAR2(64)	Information set by the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure
FIXED_TABLE_SEQUENCE	NUMBER	This contains a number that increases every time the session completes a call to the database and there has been an intervening select from a dynamic performance table. This column can be used by performance monitors to monitor statistics in the database. Each time the performance monitor looks at the database, it only needs to look at sessions that are currently active or have a higher value in this column than the highest value that the performance monitor saw the last time. All the other sessions have been idle since the last time the performance monitor looked at the database.

Column	Datatype	Description
ROW_WAIT_OBJ#	NUMBER	Object ID for the table containing the row specified in ROW_WAIT_ROW#
ROW_WAIT_FILE#	NUMBER	Identifier for the datafile containing the row specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_BLOCK#	NUMBER	Identifier for the block containing the row specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_ROW#	NUMBER	Current row being locked. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
LOGON_TIME	DATE	Time of logon
LAST_CALL_ET	NUMBER	Last call
PDML_ENABLED	VARCHAR2(3)	This column has been replaced by column PDML_STATUS
FAILOVER_TYPE	VARCHAR2(13)	Indicates whether and to what extent transparent application failover (TAF) is enabled for the session: <ul style="list-style-type: none"> ■ NONE - Failover is disabled for this session ■ SESSION - Client is able to fail over its session following a disconnect ■ SELECT - Client is able to fail over queries in progress as well See Also: <ul style="list-style-type: none"> ■ <i>Oracle Database Concepts</i> for more information on TAF ■ <i>Oracle Net Services Administrator's Guide</i> for information on configuring TAF
FAILOVER_METHOD	VARCHAR2(10)	Indicates the transparent application failover method for the session: <ul style="list-style-type: none"> ■ NONE - Failover is disabled for this session ■ BASIC - Client itself reconnects following a disconnect ■ PRECONNECT - Backup instance can support all connections from every instance for which it is backed up
FAILED_OVER	VARCHAR2(3)	Indicates whether the session is running in failover mode and failover has occurred (YES) or not (NO)
RESOURCE_CONSUMER_GROUP	VARCHAR2(32)	Name of the session's current resource consumer group
PDML_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DML enabled mode. If DISABLED, PARALLEL DML enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DML.
PDDL_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DDL enabled mode. If DISABLED, PARALLEL DDL enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DDL.
PQ_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL QUERY enabled mode. If DISABLED, PARALLEL QUERY enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL QUERY.
CURRENT_QUEUE_DURATION	NUMBER	If queued (1), the current amount of time the session has been queued. If not currently queued, the value is 0.
CLIENT_IDENTIFIER	VARCHAR2(64)	Client identifier of the session
BLOCKING_SESSION_STATUS	VARCHAR2(13)	Blocking session status: <ul style="list-style-type: none"> ■ VALID ■ NO HOLDER ■ GLOBAL ■ UNIMPLEMENTED ■ UNKNOWN
BLOCKING_SESSION	NUMBER	Session identifier of blocking session

Column	Datatype	Description
SEQ#	NUMBER	Sequence number that uniquely identifies the wait. Incremented for each wait.
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting See Also: Appendix C, "Oracle Wait Events"
P1TEXT	VARCHAR2(64)	Description of the first additional parameter
P1	NUMBER	First additional parameter
P1RAW	RAW(4)	First additional parameter
P2TEXT	VARCHAR2(64)	Description of the second additional parameter
P2	NUMBER	Second additional parameter
P2RAW	RAW(4)	Second additional parameter
P3TEXT	VARCHAR2(64)	Description of the third additional parameter
P3	NUMBER	Third additional parameter
P3RAW	RAW(4)	Third additional parameter
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
WAIT_TIME	NUMBER	A nonzero value is the session's last wait time. A zero value means the session is currently waiting.
SECONDS_IN_WAIT	NUMBER	If WAIT_TIME = 0, then SECONDS_IN_WAIT is the seconds spent in the current wait condition. If WAIT_TIME > 0, then SECONDS_IN_WAIT is the seconds since the start of the last wait, and SECONDS_IN_WAIT - WAIT_TIME / 100 is the active seconds since the last wait ended.
STATE	VARCHAR2(19)	Wait state: <ul style="list-style-type: none"> ■ 0 - WAITING (the session is currently waiting) ■ -2 - WAITED UNKNOWN TIME (duration of last wait is unknown) ■ -1 - WAITED SHORT TIME (last wait <1/100th of a second) ■ >0 - WAITED KNOWN TIME (WAIT_TIME = duration of last wait)
SERVICE_NAME	VARCHAR2(64)	Service name of the session

Table 5-5 COMMAND Column of V\$SESSION and Corresponding Commands

Number	Command	Number	Command
1	CREATE TABLE	2	INSERT
3	SELECT	4	CREATE CLUSTER
5	ALTER CLUSTER	6	UPDATE
7	DELETE	8	DROP CLUSTER
9	CREATE INDEX	10	DROP INDEX
11	ALTER INDEX	12	DROP TABLE
13	CREATE SEQUENCE	14	ALTER SEQUENCE
15	ALTER TABLE	16	DROP SEQUENCE
17	GRANT OBJECT	18	REVOKE OBJECT
19	CREATE SYNONYM	20	DROP SYNONYM
21	CREATE VIEW	22	DROP VIEW
23	VALIDATE INDEX	24	CREATE PROCEDURE
25	ALTER PROCEDURE	26	LOCK

Table 5–5 (Cont.) COMMAND Column of V\$SESSION and Corresponding Commands

Number	Command	Number	Command
27	NO-OP	28	RENAME
29	COMMENT	30	AUDIT OBJECT
31	NOAUDIT OBJECT	32	CREATE DATABASE LINK
33	DROP DATABASE LINK	34	CREATE DATABASE
35	ALTER DATABASE	36	CREATE ROLLBACK SEG
37	ALTER ROLLBACK SEG	38	DROP ROLLBACK SEG
39	CREATE TABLESPACE	40	ALTER TABLESPACE
41	DROP TABLESPACE	42	ALTER SESSION
43	ALTER USER	44	COMMIT
45	ROLLBACK	46	SAVEPOINT
47	PL/SQL EXECUTE	48	SET TRANSACTION
49	ALTER SYSTEM	50	EXPLAIN
51	CREATE USER	52	CREATE ROLE
53	DROP USER	54	DROP ROLE
55	SET ROLE	56	CREATE SCHEMA
57	CREATE CONTROL FILE	59	CREATE TRIGGER
60	ALTER TRIGGER	61	DROP TRIGGER
62	ANALYZE TABLE	63	ANALYZE INDEX
64	ANALYZE CLUSTER	65	CREATE PROFILE
66	DROP PROFILE	67	ALTER PROFILE
68	DROP PROCEDURE	70	ALTER RESOURCE COST
71	CREATE MATERIALIZED VIEW LOG	72	ALTER MATERIALIZED VIEW LOG
73	DROP MATERIALIZED VIEW LOG	74	CREATE MATERIALIZED VIEW
75	ALTER MATERIALIZED VIEW	76	DROP MATERIALIZED VIEW
77	CREATE TYPE	78	DROP TYPE
79	ALTER ROLE	80	ALTER TYPE
81	CREATE TYPE BODY	82	ALTER TYPE BODY
83	DROP TYPE BODY	84	DROP LIBRARY
85	TRUNCATE TABLE	86	TRUNCATE CLUSTER
91	CREATE FUNCTION	92	ALTER FUNCTION
93	DROP FUNCTION	94	CREATE PACKAGE
95	ALTER PACKAGE	96	DROP PACKAGE
97	CREATE PACKAGE BODY	98	ALTER PACKAGE BODY
99	DROP PACKAGE BODY	100	LOGON
101	LOGOFF	102	LOGOFF BY CLEANUP
103	SESSION REC	104	SYSTEM AUDIT
105	SYSTEM NOAUDIT	106	AUDIT DEFAULT
107	NOAUDIT DEFAULT	108	SYSTEM GRANT
109	SYSTEM REVOKE	110	CREATE PUBLIC SYNONYM
111	DROP PUBLIC SYNONYM	112	CREATE PUBLIC DATABASE LINK
113	DROP PUBLIC DATABASE LINK	114	GRANT ROLE
115	REVOKE ROLE	116	EXECUTE PROCEDURE

Table 5–5 (Cont.) COMMAND Column of V\$SESSION and Corresponding Commands

Number	Command	Number	Command
117	USER COMMENT	118	ENABLE TRIGGER
119	DISABLE TRIGGER	120	ENABLE ALL TRIGGERS
121	DISABLE ALL TRIGGERS	122	NETWORK ERROR
123	EXECUTE TYPE	157	CREATE DIRECTORY
158	DROP DIRECTORY	159	CREATE LIBRARY
160	CREATE JAVA	161	ALTER JAVA
162	DROP JAVA	163	CREATE OPERATOR
164	CREATE INDEXTYPE	165	DROP INDEXTYPE
167	DROP OPERATOR	168	ASSOCIATE STATISTICS
169	DISASSOCIATE STATISTICS	170	CALL METHOD
171	CREATE SUMMARY	172	ALTER SUMMARY
173	DROP SUMMARY	174	CREATE DIMENSION
175	ALTER DIMENSION	176	DROP DIMENSION
177	CREATE CONTEXT	178	DROP CONTEXT
179	ALTER OUTLINE	180	CREATE OUTLINE
181	DROP OUTLINE	182	UPDATE INDEXES
183	ALTER OPERATOR		

V\$SESSION_CONNECT_INFO

This view displays information about network connections for the current session.

Column	Datatype	Description
SID	NUMBER	Session identifier (can be used to join this view with V\$SESSION)
AUTHENTICATION_TYPE	VARCHAR2(26)	How the user was authenticated: DATABASE - Username/password authentication OS - Operating system external user authentication NETWORK - Network protocol or ANO authentication PROXY - OCI proxy connection authentication SERVER PASSWORD EXTERNAL ADAPTERS INTERNAL GLOBAL EXTERNAL PASSWORD BASED GLOBAL USER
OSUSER	VARCHAR2(30)	External username for this database user
NETWORK_SERVICE_BANNER	VARCHAR2(4000)	Product banners for each Oracle Net service used for this connection (one row per banner)

V\$SESSION_CURSOR_CACHE

This view displays information on cursor usage for the current session.

Note: The V\$SESSION_CURSOR_CACHE view is not a measure of the effectiveness of the SESSION_CACHED_CURSORS initialization parameter.

Column	Datatype	Description
MAXIMUM	NUMBER	Maximum number of cursors to cache. Once you hit this number, some cursors will need to be closed in order to open more. The value in this column is derived from the initialization parameter OPEN_CURSORS.
COUNT	NUMBER	The current number of cursors (whether they are in use or not)
OPENED_ONCE	NUMBER	Number of cursors opened at least once
OPEN	NUMBER	Current number of open cursors
OPENS	NUMBER	Cumulative total of cursor opens minus one. This is because the cursor that is currently open and being used for this query is not counted in the OPENS statistic.
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times an open cursor was found divided by the number of times a cursor was sought

V\$SESSION_EVENT

This view lists information on waits for an event by a session. Note that the TIME_WAITED and AVERAGE_WAIT columns will contain a value of zero on those platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set TIMED_STATISTICS to true in the parameter file. Please remember that doing this will have a small negative effect on system performance.

See Also: ["TIMED_STATISTICS"](#) on page 1-139

Column	Datatype	Description
SID	NUMBER	ID of the session
EVENT	VARCHAR2 (64)	Name of the wait event See Also: Appendix C, "Oracle Wait Events"
TOTAL_WAITS	NUMBER	Total number of waits for the event by the session
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event by the session
TIME_WAITED	NUMBER	Total amount of time waited for the event by the session (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event by the session (in hundredths of a second)
MAX_WAIT	NUMBER	Maximum time waited for the event by the session (in hundredths of a second)
TIME_WAITED_MICRO	NUMBER	Total amount of time waited for the event by the session (in microseconds)
EVENT_ID	NUMBER	Identifier of the wait event

V\$SESSION_LONGOPS

This view displays the status of various operations that run for longer than 6 seconds (in absolute time). These operations currently include many backup and recovery functions, statistics gathering, and query execution, and more operations are added for every Oracle release.

To monitor query execution progress, you must be using the cost-based optimizer and you must:

- Set the `TIMED_STATISTICS` or `SQL_TRACE` parameter to `true`
- Gather statistics for your objects with the `ANALYZE` statement or the `DBMS_STATS` package

You can add information to this view about application-specific long-running operations by using the `DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS` procedure.

See Also: *PL/SQL Packages and Types Reference* for more information on `DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS`

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
OPNAME	VARCHAR2(64)	Brief description of the operation
TARGET	VARCHAR2(64)	The object on which the operation is carried out
TARGET_DESC	VARCHAR2(32)	Description of the target
SOFAR	NUMBER	The units of work done so far
TOTALWORK	NUMBER	The total units of work
UNITS	VARCHAR2(32)	The units of measurement
START_TIME	DATE	The starting time of operation
LAST_UPDATE_TIME	DATE	Time when statistics last updated
TIME_REMAINING	NUMBER	Estimate (in seconds) of time remaining for the operation to complete
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds from the start of operations
CONTEXT	NUMBER	Context
MESSAGE	VARCHAR2(512)	Statistics summary message
USERNAME	VARCHAR2(30)	User ID of the user performing the operation
SQL_ADDRESS	RAW(4 8)	Used with the value of the <code>SQL_HASH_VALUE</code> column to identify the SQL statement associated with the operation
SQL_HASH_VALUE	NUMBER	Used with the value of the <code>SQL_ADDRESS</code> column to identify the SQL statement associated with the operation
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement associated with the operation
QCSID	NUMBER	Session identifier of the parallel coordinator

V\$SESSION_OBJECT_CACHE

This view displays object cache statistics for the current user session on the local server (instance).

Column	Datatype	Description
PINS	NUMBER	Number of object pins or look-ups in the cache
HITS	NUMBER	Number of object pins that found the object already in the cache
TRUE_HITS	NUMBER	Number of object pins that found the object already in the cache and in the desired state (thus, not requiring refresh from the database)
HIT_RATIO	NUMBER	The ratio of <code>HITS/PINS</code>
TRUE_HIT_RATIO	NUMBER	The ratio of <code>TRUE_HITS/PINS</code>

Column	Datatype	Description
OBJECT_REFRESHES	NUMBER	Number of objects in the cache that were refreshed with a new value from the database
CACHE_REFRESHES	NUMBER	Number of times the whole cache (all objects) were refreshed
OBJECT_FLUSHES	NUMBER	Number of objects in the cache that were flushed to the database
CACHE_FLUSHES	NUMBER	Number of times the whole cache (all objects) were flushed to the database
CACHE_SHRINKS	NUMBER	Number of times the cache was shrunk to the optimal size
CACHED_OBJECTS	NUMBER	Number of objects currently cached
PINNED_OBJECTS	NUMBER	Number of objects currently pinned
CACHE_SIZE	NUMBER	Current size of the cache in bytes
OPTIMAL_SIZE	NUMBER	Optimal size of the cache in bytes
MAXIMUM_SIZE	NUMBER	Maximum size of the cache in bytes

V\$SESSION_WAIT

V\$SESSION_WAIT displays the resources or events for which active sessions are waiting.

The following are tuning considerations:

- P1RAW, P2RAW, and P3RAW display the same values as the P1, P2, and P3 columns, except that the numbers are displayed in hexadecimal.
- The WAIT_TIME column contains a value of -2 on platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, then you must set the TIMED_STATISTICS initialization parameter to true. Remember that doing this has a small negative effect on system performance.

In previous releases, the WAIT_TIME column contained an arbitrarily large value instead of a negative value to indicate the platform did not have a fast timing mechanism.

- The STATE column interprets the value of WAIT_TIME and describes the state of the current or most recent wait.

Column	Datatype	Description
SID	NUMBER	Session identifier
SEQ#	NUMBER	Sequence number that uniquely identifies this wait. Incremented for each wait.
EVENT	VARCHAR2 (64)	Resource or event for which the session is waiting See Also: Appendix C, "Oracle Wait Events"
P1TEXT	VARCHAR2 (64)	Description of the first additional parameter
P1	NUMBER	First additional parameter
P1RAW	RAW (4)	First additional parameter
P2TEXT	VARCHAR2 (64)	Description of the second additional parameter
P2	NUMBER	Second additional parameter
P2RAW	RAW (4)	Second additional parameter
P3TEXT	VARCHAR2 (64)	Description of the third additional parameter
P3	NUMBER	Third additional parameter
P3RAW	RAW (4)	Third additional parameter

V\$SESSION_WAIT_CLASS

Column	Datatype	Description
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
WAIT_TIME	NUMBER	A nonzero value is the session's last wait time. A zero value means the session is currently waiting.
SECONDS_IN_WAIT	NUMBER	If WAIT_TIME = 0, then SECONDS_IN_WAIT is the seconds spent in the current wait condition. If WAIT_TIME > 0, then SECONDS_IN_WAIT is the seconds since the start of the last wait, and SECONDS_IN_WAIT - WAIT_TIME / 100 is the active seconds since the last wait ended.
STATE	VARCHAR2(19)	Wait state: <ul style="list-style-type: none">■ 0 - WAITING (the session is currently waiting)■ -2 - WAITED UNKNOWN TIME (duration of last wait is unknown)■ -1 - WAITED SHORT TIME (last wait <1/100th of a second)■ >0 - WAITED KNOWN TIME (WAIT_TIME = duration of last wait)

See Also: ["TIMED_STATISTICS"](#) on page 1-139 and [Appendix C, "Oracle Wait Events"](#)

V\$SESSION_WAIT_CLASS

V\$SESSION_WAIT_CLASS displays the time spent in various wait event operations on a per-session basis.

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Serial number
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
TOTAL_WAITS	NUMBER	Number of times waits of the class occurred for the session
TIME_WAITED	NUMBER	Amount of time spent in the wait class by the session

V\$SESSION_WAIT_HISTORY

V\$SESSION_WAIT_HISTORY displays the last 10 wait events for each active session.

Column	Datatype	Description
SID	NUMBER	Session identifier
SEQ#	NUMBER	Sequence of wait events; 1 is the most recent
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting
P1TEXT	VARCHAR2(64)	Description of the first additional parameter
P1	NUMBER	First additional parameter
P2TEXT	VARCHAR2(64)	Description of the second additional parameter
P2	NUMBER	Second additional parameter
P3TEXT	VARCHAR2(64)	Description of the third additional parameter
P3	NUMBER	Third additional parameter

Column	Datatype	Description
WAIT_TIME	NUMBER	A nonzero value is the session's last wait time. A zero value means the session is currently waiting.

V\$SESSMETRIC

V\$SESSMETRIC displays the metric values for all sessions.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
SESSION_ID	NUMBER	Session ID
SESSION_SERIAL_NUM	NUMBER	Session serial number
CPU	NUMBER	CPU usage
PHYSICAL_READS	NUMBER	Number of physical reads
PGA_MEMORY	NUMBER	PGA size at the end of the interval
HARD_PARSSES	NUMBER	Number of hard parses
SOFT_PARSSES	NUMBER	Number of soft parses
PHYSICAL_READ_PCT	NUMBER	Physical read ratio
LOGICAL_READ_PCT	NUMBER	Logical read ratio

V\$SESSTAT

This view lists user session statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V\$STATNAME view.

Column	Datatype	Description
SID	NUMBER	Session identifier
STATISTIC#	NUMBER	Statistic number Note: Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
VALUE	NUMBER	Statistic value

See Also: "[V\\$STATNAME](#)" on page 5-71 and [Appendix E, "Statistics Descriptions"](#)

V\$SGA

V\$SGA displays summary information about the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2(20)	SGA component group
VALUE	NUMBER	Memory size (in bytes)

V\$SGA_CURRENT_RESIZE_OPS

V\$SGA_CURRENT_RESIZE_OPS displays information about SGA resize operations which are currently in progress. An operation can be a grow or a shrink of a dynamic SGA component. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: <ul style="list-style-type: none"> ▪ STATIC ▪ INITIALIZING ▪ DISABLED ▪ GROW ▪ SHRINK ▪ SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: <ul style="list-style-type: none"> ▪ MANUAL ▪ DEFERRED ▪ IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Desired value of the parameter after the resize
CURRENT_SIZE	NUMBER	Current value of the parameter
START_TIME	DATE	Start time of the operation
LAST_UPDATE_TIME	DATE	Last time progress was made for the operation

V\$SGA_DYNAMIC_COMPONENTS

V\$SGA_DYNAMIC_COMPONENTS displays information about the dynamic SGA components. This view summarizes information based on all completed SGA resize operations since instance startup. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
CURRENT_SIZE	NUMBER	Current size of the component
MIN_SIZE	NUMBER	Minimum size of the component since instance startup
MAX_SIZE	NUMBER	Maximum size of the component since instance startup
USER_SPECIFIED_SIZE	NUMBER	Value of the user parameter for the component
OPER_COUNT	NUMBER	Number of operations since instance startup
LAST_OPER_TYPE	VARCHAR2(13)	Last completed operation for the component: <ul style="list-style-type: none"> ▪ STATIC ▪ INITIALIZING ▪ DISABLED ▪ GROW ▪ SHRINK ▪ SHRINK_CANCEL

Column	Datatype	Description
LAST_OPER_MODE	VARCHAR2(9)	Mode of the last completed operation: <ul style="list-style-type: none"> ▪ MANUAL ▪ DEFERRED ▪ IMMEDIATE
LAST_OPER_TIME	DATE	Start time of the last completed operation
GRANULE_SIZE	NUMBER	Granularity of the grow or the shrink operation

V\$SGA_DYNAMIC_FREE_MEMORY

V\$SGA_DYNAMIC_FREE_MEMORY displays information about the amount of SGA memory available for future dynamic SGA resize operations.

Column	Datatype	Description
CURRENT_SIZE	NUMBER	Amount of available memory (in bytes)

V\$SGA_RESIZE_OPS

V\$SGA_RESIZE_OPS displays information about the last 400 completed SGA resize operations. This does not include in-progress operations. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: <ul style="list-style-type: none"> ▪ STATIC ▪ INITIALIZING ▪ DISABLED ▪ GROW ▪ SHRINK ▪ SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: <ul style="list-style-type: none"> ▪ MANUAL ▪ DEFERRED ▪ IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Requested value of the parameter after the resize
FINAL_SIZE	NUMBER	Real value of the parameter after the resize
STATUS	VARCHAR2(9)	Completion status of the operation: <ul style="list-style-type: none"> ▪ INACTIVE ▪ PENDING ▪ COMPLETE ▪ CANCELLED ▪ ERROR
START_TIME	DATE	Start time of the operation
END_TIME	DATE	End time of the operation

V\$SGAINFO

V\$SGAINFO displays size information about the SGA, including the sizes of different SGA components, the granule size, and free memory.

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the SGA statistic
BYTES	NUMBER	Size of the statistic (in bytes)
RESIZEABLE	VARCHAR2(3)	Indicates whether the statistic is resizeable (YES) or not (NO)

V\$SGASTAT

V\$SGASTAT displays detailed information on the system global area (SGA).

Column	Datatype	Description
POOL	VARCHAR2(12)	Designates the pool in which the memory in NAME resides: <ul style="list-style-type: none"> ▪ <code>shared pool</code> - Memory is allocated from the shared pool ▪ <code>large pool</code> - Memory is allocated from the large pool ▪ <code>java pool</code> - Memory is allocated from the Java pool ▪ <code>streams pool</code> - Memory is allocated from the Streams pool
NAME	VARCHAR2(26)	SGA component name
BYTES	NUMBER	Memory size in bytes

V\$SHARED_POOL_ADVICE

V\$SHARED_POOL_ADVICE displays information about estimated parse time in the shared pool for different pool sizes. The sizes range from 10% of the current shared pool size or the amount of pinned library cache memory (whichever is higher) to 200% of the current shared pool size, in equal intervals. The value of the interval depends on the current size of the shared pool.

Column	Datatype	Description
SHARED_POOL_SIZE_FOR_ESTIMATE	NUMBER	Shared pool size for the estimate (in megabytes)
SHARED_POOL_SIZE_FACTOR	NUMBER	Size factor with respect to the current shared pool size
ESTD_LC_SIZE	NUMBER	Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER	Estimated number of library cache memory objects in the shared pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER	Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a shared pool of the specified size. This is the time that would have been spent in reloading the required objects in the shared pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER	Estimated parse time saved factor with respect to the current shared pool size
ESTD_LC_LOAD_TIME	NUMBER	Estimated elapsed time (in seconds) for parsing in a shared pool of the specified size
ESTD_LC_LOAD_TIME_FACTOR	NUMBER	Estimated load time factor with respect to the current shared pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER	Estimated number of times a library cache memory object was found in a shared pool of the specified size

V\$SHARED_POOL_RESERVED

This fixed view lists statistics that help you tune the reserved pool and space within the shared pool.

Column	Datatype	Description
The following columns of V\$SHARED_POOL_RESERVED are valid only if the initialization parameter SHARED_POOL_RESERVED_SIZE is set to a valid value.		
See Also: "SHARED_POOL_RESERVED_SIZE" on page 1-127		
FREE_SPACE	NUMBER	Total amount of free space on the reserved list
AVG_FREE_SIZE	NUMBER	Average size of the free memory on the reserved list
FREE_COUNT	NUMBER	Number of free pieces of memory on the reserved list
MAX_FREE_SIZE	NUMBER	Size of the largest free piece of memory on the reserved list
USED_SPACE	NUMBER	Total amount of used memory on the reserved list
AVG_USED_SIZE	NUMBER	Average size of the used memory on the reserved list
USED_COUNT	NUMBER	Number of used pieces of memory on the reserved list
MAX_USED_SIZE	NUMBER	Size of the largest used piece of memory on the reserved list
REQUESTS	NUMBER	Number of times that the reserved list was searched for a free piece of memory
REQUEST_MISSES	NUMBER	Number of times the reserved list did not have a free piece of memory to satisfy the request, and started flushing objects from the LRU list
LAST_MISS_SIZE	NUMBER	Request size of the last request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list
MAX_MISS_SIZE	NUMBER	Request size of the largest request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list
The following columns of V\$SHARED_POOL_RESERVED contain values which are valid even if SHARED_POOL_RESERVED_SIZE is not set.		
REQUEST_FAILURES	NUMBER	Number of times that no memory was found to satisfy a request (that is, the number of times the error ORA-04031 occurred)
LAST_FAILURE_SIZE	NUMBER	Request size of the last failed request (that is, the request size for the last ORA-04031 error)
ABORTED_REQUEST_THRESHOLD	NUMBER	Minimum size of a request which signals an ORA-04031 error without flushing objects
ABORTED_REQUESTS	NUMBER	Number of requests that signalled an ORA-04031 error without flushing objects
LAST_ABORTED_SIZE	NUMBER	Last size of the request that returned an ORA-04031 error without flushing objects from the LRU list

V\$SHARED_SERVER

This view contains information on the shared server processes.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the server
PADDR	RAW(4 8)	Server's process address

Column	Datatype	Description
STATUS	VARCHAR2 (16)	Server status: EXEC - Executing SQL WAIT (ENQ) - Waiting for a lock WAIT (SEND) - Waiting to send data to user WAIT (COMMON) - Idle; waiting for a user request WAIT (RESET) - Waiting for a circuit to reset after a break QUIT - Terminating
MESSAGES	NUMBER	Number of messages processed
BYTES	NUMBER	Total number of bytes in all messages
BREAKS	NUMBER	Number of breaks
CIRCUIT	RAW (4 8)	Address of circuit currently being serviced
IDLE	NUMBER	Total idle time (in hundredths of a second)
BUSY	NUMBER	Total busy time (in hundredths of a second)
REQUESTS	NUMBER	Total number of requests taken from the common queue in this server's lifetime

V\$SHARED_SERVER_MONITOR

This view contains information for tuning the shared server.

Column	Datatype	Description
MAXIMUM_CONNECTIONS	NUMBER	Highest number of virtual circuits in use at one time since the instance started. If this value reaches the value set for the <code>CIRCUITS</code> initialization parameter, then consider raising the value of <code>CIRCUITS</code> . See Also: " CIRCUITS " on page 1-25
MAXIMUM_SESSIONS	NUMBER	Highest number of shared server sessions in use at one time since the instance started. If this reaches the value set for the <code>SHARED_SERVER_SESSIONS</code> initialization parameter, then consider raising the value of <code>SHARED_SERVER_SESSIONS</code> . See Also: " SHARED_SERVER_SESSIONS " on page 1-128
SERVERS_STARTED	NUMBER	Total number of shared servers started since the instance started (but not including those started during startup)
SERVERS_TERMINATED	NUMBER	Total number of shared servers stopped by Oracle since the instance started
SERVERS_HIGHWATER	NUMBER	Highest number of servers running at one time since the instance started. If this value reaches the value set for the <code>MAX_SHARED_SERVERS</code> initialization parameter, then consider raising the value of <code>SHARED_SERVERS</code> . See Also: " SHARED_SERVERS " on page 1-129

V\$SORT_SEGMENT

This view contains information about every sort segment in a given instance. The view is only updated when the tablespace is of the `TEMPORARY` type.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2 (31)	Name of tablespace
SEGMENT_FILE	NUMBER	File number of the first extent
SEGMENT_BLOCK	NUMBER	Block number of the first extent
EXTENT_SIZE	NUMBER	Extent size

Column	Datatype	Description
CURRENT_USERS	NUMBER	Number of active users of the segment
TOTAL_EXTENTS	NUMBER	Total number of extents in the segment
TOTAL_BLOCKS	NUMBER	Total number of blocks in the segment
USED_EXTENTS	NUMBER	Extents allocated to active sorts
USED_BLOCKS	NUMBER	Blocks allocated to active sorts
FREE_EXTENTS	NUMBER	Extents not allocated to any sort
FREE_BLOCKS	NUMBER	Blocks not allocated to any sort
ADDED_EXTENTS	NUMBER	Number of extent allocations
EXTENT_HITS	NUMBER	Number of times an unused extent was found in the pool
FREED_EXTENTS	NUMBER	Number of deallocated extents
FREE_REQUESTS	NUMBER	Number of requests to deallocate
MAX_SIZE	NUMBER	Maximum number of extents ever used
MAX_BLOCKS	NUMBER	Maximum number of blocks ever used
MAX_USED_SIZE	NUMBER	Maximum number of extents used by all sorts
MAX_USED_BLOCKS	NUMBER	Maximum number of blocks used by all sorts
MAX_SORT_SIZE	NUMBER	Maximum number of extents used by an individual sort
MAX_SORT_BLOCKS	NUMBER	Maximum number of blocks used by an individual sort
RELATIVE_FNO	NUMBER	Relative file number of the sort segment header

V\$SPPARAMETER

V\$SPPARAMETER displays information about the contents of the server parameter file. If a server parameter file was not used to start the instance, then each row of the view will contain FALSE in the ISSPECIFIED column.

Column	Datatype	Description
SID	VARCHAR2(80)	SID for which the parameter is defined
NAME	VARCHAR2(80)	Name of the parameter
VALUE	VARCHAR2(255)	Parameter value (null if a server parameter file was not used to start the instance)
DISPLAY_VALUE	VARCHAR2(255)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISSPECIFIED	VARCHAR2(6)	Indicates whether the parameter was specified in the server parameter file (TRUE) or not (FALSE)
ORDINAL	NUMBER	Position (ordinal number) of the parameter value (0 if a server parameter file was not used to start the instance). Useful only for parameters whose values are lists of strings.
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update (null if a server parameter file was not used to start the instance)

V\$SQL

V\$SQL lists statistics on shared SQL area without the GROUP BY clause and contains one row for each child of the original SQL text entered.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor

Column	Datatype	Description
SQL_FULLTEXT	CLOB	Full text for the SQL statement exposed as a CLOB column. The full text of a SQL statement can be retrieved using this column instead of joining with the V\$SQL_TEXT dynamic performance view.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
SHARABLE_MEM	NUMBER	Amount of shared memory used by the child cursor (in bytes)
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of the child cursor (in bytes)
RUNTIME_MEM	NUMBER	Fixed amount of memory required during the execution of the child cursor
SORTS	NUMBER	Number of sorts that were done for the child cursor
LOADED_VERSIONS	NUMBER	Indicates whether the context heap is loaded (1) or not (0)
OPEN_VERSIONS	NUMBER	Indicates whether the child cursor is locked (1) or not (0)
USERS_OPENING	NUMBER	Number of users executing the statement
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Number of executions that took place on this object since it was brought into the library cache
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
USERS_EXECUTING	NUMBER	Number of users executing the statement
LOADS	NUMBER	Number of times the object was either loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	Timestamp of the parent creation time
INVALIDATIONS	NUMBER	Number of times this child cursor has been invalidated
PARSE_CALLS	NUMBER	Number of parse calls for this child cursor
DISK_READS	NUMBER	Number of disk reads for this child cursor
DIRECT_WRITES	NUMBER	Number of direct writes for this child cursor
BUFFER_GETS	NUMBER	Number of buffer gets for this child cursor
APPLICATION_WAIT_TIME	NUMBER	Application wait time
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time
USER_IO_WAIT_TIME	NUMBER	User I/O Wait Time
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time
JAVA_EXEC_TIME	NUMBER	Java execution time
ROWS_PROCESSED	NUMBER	Total number of rows the parsed SQL statement returns
COMMAND_TYPE	NUMBER	Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement is executed
OPTIMIZER_COST	NUMBER	Cost of this query given by the optimizer
OPTIMIZER_ENV	RAW(648)	Optimizer environment
OPTIMIZER_ENV_HASH_VALUE	NUMBER	Hash value for the optimizer environment
PARSING_USER_ID	NUMBER	User ID of the user who originally built this child cursor
PARSING_SCHEMA_ID	NUMBER	Schema ID that was used to originally build this child cursor
KEPT_VERSIONS	NUMBER	Indicates whether this child cursor has been marked to be kept pinned in the cache using the DBMS_SHARED_POOL package

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the handle to the parent for this cursor
TYPE_CHK_HEAP	RAW(4)	Descriptor of the type check heap for this child cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
OLD_HASH_VALUE	NUMBER	Old SQL hash value
PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
CHILD_NUMBER	NUMBER	Number of this child cursor
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	Hash value of the module listed in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	Hash value of the action listed in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction fails to serialize, producing ORA-08177 errors, per cursor
OUTLINE_CATEGORY	VARCHAR2(64)	If an outline was applied during construction of the cursor, then this column displays the category of that outline. Otherwise the column is left blank.
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
OUTLINE_SID	NUMBER	Outline session identifier
CHILD_ADDRESS	RAW(4 8)	Address of the child cursor
SQLTYPE	NUMBER	Denotes the version of the SQL language used for this statement
REMOTE	VARCHAR2(1)	Indicates whether the cursor is remote mapped (Y) or not (N)
OBJECT_STATUS	VARCHAR2(19)	Status of the cursor (VALID or INVALID)
LITERAL_HASH_VALUE	NUMBER	Hash value of the literals which are replaced with system-generated bind variables and are to be matched, when CURSOR_SHARING is used. This is not the hash value for the SQL statement. If CURSOR_SHARING is not used, then the value is 0.
LAST_LOAD_TIME	VARCHAR2(19)	Time at which the query plan (heap 6) was loaded into the library cache
IS_OBSOLETE	VARCHAR2(1)	Indicates whether the cursor has become obsolete (Y) or not (N). This can happen if the number of child cursors is too large.
CHILD_LATCH	NUMBER	Child latch number that is protecting the cursor
SQL_PROFILE	VARCHAR2(64)	SQL profile
PROGRAM_ID	NUMBER	Program identifier
PROGRAM_LINE#	NUMBER	Program line number

V\$SQL_BIND_CAPTURE

V\$SQL_BIND_CAPTURE displays information on bind variables used by SQL cursors. Each row in the view contains information for one bind variable defined in a cursor. This includes:

- Reference to the cursor defining the bind variable
(hash_value, address) for the parent cursor and (hash_value, child_address) for the child cursor.

- Bind metadata
Name, position, datatype, character set ID, precision, scale, and maximum length of the bind variable.
- Bind data
One of the bind values used for the bind variable during a past execution of its associated SQL statement. Bind values are not always captured for this view. Bind values are displayed by this view only when the type of the bind variable is simple (this excludes LONG, LOB, and ADT datatypes) and when the bind variable is used in the WHERE or HAVING clauses of the SQL statement.

Bind capture is disabled when the STATISTICS_LEVEL initialization parameter is set to BASIC. This view can be joined with V\$SQLAREA on (HASH_VALUE, ADDRESS) and with V\$SQL on (HASH_VALUE, CHILD_ADDRESS).

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the parent cursor
HASH_VALUE	NUMBER	Hash value of the parent cursor in the library cache. The hash value is a fixed index for the view and should always be used to speed up the access to the view.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
CHILD_ADDRESS	RAW(4 8)	Address of the parent cursor
CHILD_NUMBER	NUMBER	Child cursor number
NAME	VARCHAR2(30)	Name of the bind variable
POSITION	NUMBER	Position of the bind variable in the SQL statement
DUP_POSITION	NUMBER	If the binding is performed by name and the bind variable is duplicated, then this column gives the position of the primary bind variable.
DATATYPE	NUMBER	Internal identifier for the bind datatype
DATATYPE_STRING	VARCHAR2(15)	Textual representation of the bind datatype
CHARACTER_SID	NUMBER	National character set identifier
PRECISION	NUMBER	Precision (for numeric binds)
SCALE	NUMBER	Scale (for numeric binds)
MAX_LENGTH	NUMBER	Maximum bind length
WAS_CAPTURED	VARCHAR2(3)	Indicates whether the bind value was captured (YES) or not (NO)
LAST_CAPTURED	DATE	Date when the bind value was captured. Bind values are captured when SQL statements are executed. To limit the overhead, binds are captured at most every 15 minutes for a given cursor.
VALUE_STRING	VARCHAR2(4000)	Value of the bind represented as a string
VALUE_ANYDATA	ANYDATA	Value of the bind represented using the self-descriptive Sys.AnyData datatype. This representation is useful to programmatically decode the value of the bind variable.

V\$SQL_BIND_DATA

For each distinct bind variable in each cursor owned by the session querying this view, this view describes:

- Actual bind data, if the bind variable is user defined
- The underlying literal, if the CURSOR_SHARING parameter is set to FORCE and the bind variable is system generated. (System-generated binds have a value of 256 in the SHARED_FLAG2 column.)

Column	Datatype	Description
CURSOR_NUM	NUMBER	Cursor number for this bind
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
SHARED_MAX_LEN	NUMBER	Shared maximum length for this bind from the shared cursor object associated with this bind
PRIVATE_MAX_LEN	NUMBER	Private maximum length for this bind sent from the client
ARRAY_SIZE	NUMBER	Maximum number of array elements (for array binds only)
PRECISION	NUMBER	Precision (for numeric binds)
SCALE	NUMBER	Scale (for numeric binds)
SHARED_FLAG	NUMBER	Shared bind data flags
SHARED_FLAG2	NUMBER	Shared bind data flags (continued)
BUF_ADDRESS	RAW(4 8)	Bind buffer memory address
BUF_LENGTH	NUMBER	Bind buffer length
VAL_LENGTH	NUMBER	Actual bind value length
BUF_FLAG	NUMBER	Bind buffer flags
INDICATOR	NUMBER	Bind indicator
VALUE	VARCHAR2(4000)	Contents of the bind buffer

V\$SQL_BIND_METADATA

For each distinct bind variable in each cursor owned by the session querying this view, this view describes:

- Bind metadata provided by the client, if the bind variable is user defined
- Metadata based on the underlying literal, if the `CURSOR_SHARING` parameter is set to `FORCE` and the bind variable is system-generated.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Memory address of the child cursor that owns this bind variable
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
MAX_LENGTH	NUMBER	Maximum length of the bind value
ARRAY_LEN	NUMBER	Maximum number of array elements (for array binds only)
BIND_NAME	VARCHAR2(30)	User-defined or system-generated bind variable name (if used)

V\$SQL_CURSOR

This view displays debugging information for each cursor associated with the session querying this view.

Column	Datatype	Description
CURNO	NUMBER	Cursor number
FLAG	NUMBER	Flags set in the cursor
STATUS	VARCHAR2(9)	Status of the cursor; that is, what state the cursor is in
PARENT_HANDLE	RAW(4 8)	Pointer to the parent cursor handle
PARENT_LOCK	RAW(4 8)	Pointer to the parent cursor lock

Column	Datatype	Description
CHILD_LOCK	RAW(4 8)	Pointer to the child cursor lock
CHILD_PIN	RAW(4 8)	Pointer to the child cursor pin
PERS_HEAP_MEM	NUMBER	Total amount of memory allocated from persistent heap for this cursor
WORK_HEAP_MEM	NUMBER	Total amount of memory allocated from the work heap for this cursor
BIND_VARS	NUMBER	Total number of bind positions in the query currently parsed into this cursor
DEFINE_VARS	NUMBER	Total number of define variables in the query currently parsed into this cursor
BIND_MEM_LOC	VARCHAR2(64)	Which memory heap the bind variables are stored in: either the UGA or the CGA
INST_FLAG	VARCHAR2(64)	Instantiation object flags
INST_FLAG2	VARCHAR2(64)	Instantiation object flags (continued)

V\$SQL_OPTIMIZER_ENV

V\$SQL_OPTIMIZER_ENV displays the contents of the optimizer environment used to build the execution plan of a SQL cursor. This view can be joined with V\$SQLAREA on (HASH_VALUE, ADDRESS) and with V\$SQL on (HASH_VALUE, CHILD_ADDRESS).

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the parent cursor
HASH_VALUE	NUMBER	Hash value of the parent cursor in the library cache. The hash value is the fixed index for the view and should be used when querying V\$SQL_OPTIMIZER_ENV to avoid scanning the entire library cache.
SQL_ID	VARCHAR2(13)	SQL identifier
CHILD_ADDRESS	RAW(4 8)	Address of the child cursor
CHILD_NUMBER	NUMBER	Child cursor number
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2(40)	Name of the parameter
ISDEFAULT	VARCHAR2(3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2(25)	Value of the parameter

V\$SQL_PLAN

V\$SQL_PLAN contains the execution plan information for each child cursor loaded in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V\$SQLAREA to add the cursor-specific information.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V\$SQL to add the child cursor-specific information.

Column	Datatype	Description
OPERATION	VARCHAR2(30)	Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)	A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(40)	Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER	Object number of the table or the index
OBJECT_OWNER	VARCHAR2(30)	Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)	Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)	Alias for the object
OBJECT_TYPE	VARCHAR2(20)	Type of the object
OPTIMIZER	VARCHAR2(20)	Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
ID	NUMBER	A number assigned to each step in the execution plan
PARENT_ID	NUMBER	ID of the next execution step that operates on the output of the current step
DEPTH	NUMBER	Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER	Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER	Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER	Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER	Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER	Estimate, by the cost-based optimizer, of the number of bytes produced by the operation
OTHER_TAG	VARCHAR2(35)	Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)	Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)	Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER	Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)	Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)	Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER	CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER	I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER	Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.

Column	Datatype	Description
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)	Expressions produced by the operation
TIME	NUMBER	Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)	Name of the query block
REMARKS	VARCHAR2(4000)	Remarks

V\$SQL_PLAN_STATISTICS

V\$SQL_PLAN_STATISTICS provides execution statistics at the row source level for each child cursor.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V\$SQLAREA to locate the parent cursor.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache
PLAN_HASH_VALUE	NUMBER	Plan hash value
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V\$SQL to locate the child cursor using this area.
OPERATION_ID	NUMBER	A number assigned to each step in the execution plan
EXECUTIONS	NUMBER	Number of times this cursor has been executed
LAST_STARTS	NUMBER	Number of times this operation has been started, during the last execution
STARTS	NUMBER	Number of times this operation has been started, accumulated over the past executions
LAST_OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, during the last execution
OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, accumulated over the past executions
LAST_CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, during the last execution. Buffers are usually retrieved in consistent mode for queries.
CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, accumulated over the past executions. Buffers are usually retrieved in consistent mode for queries.
LAST_CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, during the last execution. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, accumulated over the past executions. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
LAST_DISK_READS	NUMBER	Number of physical disk reads performed by the operation, during the last execution
DISK_READS	NUMBER	Number of physical disk reads performed by the operation, accumulated over the past executions
LAST_DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, during the last execution
DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, accumulated over the past executions
LAST_ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, during the last execution

Column	Datatype	Description
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, accumulated over the past executions

V\$SQL_PLAN_STATISTICS_ALL

V\$SQL_PLAN_STATISTICS_ALL contains memory usage statistics for row sources that use SQL memory (sort or hash-join). This view concatenates information in V\$SQL_PLAN with execution statistics from V\$SQL_PLAN_STATISTICS and V\$SQL_WORKAREA.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V\$SQLAREA to add the cursor-specific information.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache
PLAN_HASH_VALUE	NUMBER	Plan hash value
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V\$SQL to add the child cursor-specific information.
OPERATION	VARCHAR2(30)	Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)	A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(40)	Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER	Object number of the table or the index
OBJECT_OWNER	VARCHAR2(30)	Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)	Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)	Alias for the object
OBJECT_TYPE	VARCHAR2(20)	Type of the object
OPTIMIZER	VARCHAR2(20)	Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
ID	NUMBER	A number assigned to each step in the execution plan
PARENT_ID	NUMBER	ID of the next execution step that operates on the output of the current step
DEPTH	NUMBER	Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER	Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER	Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER	Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER	Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER	Estimate, by the cost-based optimizer, of the number of bytes produced by the operation

Column	Datatype	Description
OTHER_TAG	VARCHAR2(35)	Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)	Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)	Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER	Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)	Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)	Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER	CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER	I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER	Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)	Expressions produced by the operation
TIME	NUMBER	Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)	Name of the query block
REMARKS	VARCHAR2(4000)	Remarks
EXECUTIONS	NUMBER	Number of times this cursor has been executed
LAST_STARTS	NUMBER	Number of times this operation has been started, during the last execution
STARTS	NUMBER	Number of times this operation has been started, accumulated over the past executions
LAST_OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, during the last execution
OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, accumulated over the past executions
LAST_CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, during the last execution. Buffers are usually retrieved in consistent mode for queries.
CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, accumulated over the past executions. Buffers are usually retrieved in consistent mode for queries.
LAST_CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, during the last execution. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, accumulated over the past executions. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
LAST_DISK_READS	NUMBER	Number of physical disk reads performed by the operation, during the last execution
DISK_READS	NUMBER	Number of physical disk reads performed by the operation, accumulated over the past executions
LAST_DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, during the last execution

Column	Datatype	Description
DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, accumulated over the past executions
LAST_ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, during the last execution
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, accumulated over the past executions
POLICY	VARCHAR2(10)	Sizing policy for this work area: <ul style="list-style-type: none"> ■ MANUAL ■ AUTO
ESTIMATED_OPTIMAL_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation completely in memory (optimal execution). This is either derived from optimizer statistics or from previous executions.
ESTIMATED_ONEPASS_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation in a single pass. This is either derived from optimizer statistics or from previous executions.
LAST_MEMORY_USED	NUMBER	Memory size (in KB) used by this work area during the last execution of the cursor
LAST_EXECUTION	VARCHAR2(10)	Indicates whether this work area ran using OPTIMAL, ONE PASS, or under ONE PASS memory requirement (MULTI-PASS), during the last execution of the cursor
LAST_DEGREE	NUMBER	Degree of parallelism used, during the last execution of the cursor
TOTAL_EXECUTIONS	NUMBER	Number of times this work area was active
OPTIMAL_EXECUTIONS	NUMBER	Number of times this work area ran in optimal mode
ONEPASS_EXECUTIONS	NUMBER	Number of times this work area ran in one pass mode
MULTIPASSES_EXECUTIONS	NUMBER	Number of times this work area ran below the one pass memory requirement
ACTIVE_TIME	NUMBER	Average time this work area is active (in hundredths of a second)
MAX_TEMPSEG_SIZE	NUMBER	Maximum temporary segment size (in bytes) created by an instantiation of this work area. This column is null if this work area has never spilled to disk.
LAST_TEMPSEG_SIZE	NUMBER	Temporary segment size (in bytes) created in the last instantiation of this work area. This column is null if the last instantiation of this work area did not spill to disk.

V\$SQL_REDIRECTION

This view identifies SQL statements that are redirected.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the cursor handle
PARENT_HANDLE	RAW(4 8)	Address of the parent cursor handle
HASH_VALUE	NUMBER	Hash value of the SQL statement
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement
CHILD_NUMBER	NUMBER	Number of the child (instance) for the hash
PARSING_USER_ID	NUMBER	Parsing user ID
PARSING_SCHEMA_ID	NUMBER	Parsing schema ID
COMMAND_TYPE	NUMBER	SELECT, UPDATE, INSERT, MERGE
REASON	VARCHAR2(14)	Reason for redirection ('INVALID OBJECT', 'ROWID', 'QUERY REWRITE', 'READ ONLY')
ERROR_CODE	NUMBER	Error code for local parse

Column	Datatype	Description
POSITION	NUMBER	Error position, 0 if unknown
SQL_TEXT_PIECE	VARCHAR2(1000)	SQL Text containing position, usually a (qualified) identifier
ERROR_MESSAGE	VARCHAR2(1000)	Error code's corresponding error message resolved in the database language, no arguments resolved

V\$SQL_SHARED_CURSOR

This view explains why a particular child cursor is not shared with existing child cursors. Each column identifies a specific reason why the cursor cannot be shared.

Column	Datatype	Description
SQL_ID	VARCHAR2(13)	SQL identifier
ADDRESS	RAW(4 8)	Address of the parent cursor
CHILD_ADDRESS	RAW(4 8)	Address of the child cursor
CHILD_NUMBER	NUMBER	Child number
UNBOUND_CURSOR	VARCHAR2(1)	(Y N) The existing child cursor was not fully built (in other words, it was not optimized)
SQL_TYPE_MISMATCH	VARCHAR2(1)	(Y N) The SQL type does not match the existing child cursor
OPTIMIZER_MISMATCH	VARCHAR2(1)	(Y N) The optimizer environment does not match the existing child cursor
OUTLINE_MISMATCH	VARCHAR2(1)	(Y N) The outlines do not match the existing child cursor
STATS_ROW_MISMATCH	VARCHAR2(1)	(Y N) The existing statistics do not match the existing child cursor
LITERAL_MISMATCH	VARCHAR2(1)	(Y N) Non-data literal values do not match the existing child cursor
SEC_DEPTH_MISMATCH	VARCHAR2(1)	(Y N) Security level does not match the existing child cursor
EXPLAIN_PLAN_CURSOR	VARCHAR2(1)	(Y N) The child cursor is an explain plan cursor and should not be shared
BUFFERED_DML_MISMATCH	VARCHAR2(1)	(Y N) Buffered DML does not match the existing child cursor
PDML_ENV_MISMATCH	VARCHAR2(1)	(Y N) PDML environment does not match the existing child cursor
INST_DRTLD_MISMATCH	VARCHAR2(1)	(Y N) Insert direct load does not match the existing child cursor
SLAVE_QC_MISMATCH	VARCHAR2(1)	(Y N) The existing child cursor is a slave cursor and the new one was issued by the coordinator (or, the existing child cursor was issued by the coordinator and the new one is a slave cursor)
TYPECHECK_MISMATCH	VARCHAR2(1)	(Y N) The existing child cursor is not fully optimized
AUTH_CHECK_MISMATCH	VARCHAR2(1)	(Y N) Authorization/translation check failed for the existing child cursor
BIND_MISMATCH	VARCHAR2(1)	(Y N) The bind metadata does not match the existing child cursor
DESCRIBE_MISMATCH	VARCHAR2(1)	(Y N) The typecheck heap is not present during the describe for the child cursor
LANGUAGE_MISMATCH	VARCHAR2(1)	(Y N) The language handle does not match the existing child cursor
TRANSLATION_MISMATCH	VARCHAR2(1)	(Y N) The base objects of the existing child cursor do not match
ROW_LEVEL_SEC_MISMATCH	VARCHAR2(1)	(Y N) The row level security policies do not match
INSUFF_PRIVS	VARCHAR2(1)	(Y N) Insufficient privileges on objects referenced by the existing child cursor
INSUFF_PRIVS_REM	VARCHAR2(1)	(Y N) Insufficient privileges on remote objects referenced by the existing child cursor
REMOTE_TRANS_MISMATCH	VARCHAR2(1)	(Y N) The remote base objects of the existing child cursor do not match
LOGMINER_SESSION_MISMATCH	VARCHAR2(1)	(Y N)
INCOMP_LTRL_MISMATCH	VARCHAR2(1)	(Y N)

V\$SQL_SHARED_MEMORY

This view displays information about the cursor shared memory snapshot. Each SQL statement stored in the shared pool has one or more child objects associated with it. Each child object has a number of parts, one of which is the context heap, which holds, among other things, the query plan.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	SQL text of the shared cursor child object for which this row is displaying information
SQL_FULLTEXT	CLOB	Full text for the SQL statement exposed as a CLOB column. The full text of a SQL statement can be retrieved using this column instead of joining with the V\$SQL_TEXT dynamic performance view.
HASH_VALUE	NUMBER	Hash value of the above SQL text in the shared pool
SQL_ID	VARCHAR2(13)	SQL identifier of the above SQL text in the shared pool
HEAP_DESC	RAW(4 8)	Address of the descriptor for the context heap of the child cursor described in this row
STRUCTURE	VARCHAR2(16)	If the memory chunk described in this row was allocated using a comment of the form "X: Y", then this is the "X" part of the comment
FUNCTION	VARCHAR2(16)	Similar to the STRUCTURE column, this is the "Y" field of the comment
CHUNK_COM	VARCHAR2(16)	Whole comment field that was supplied when this memory chunk was allocated
CHUNK_PTR	RAW(4 8)	Starting address of the allocated memory chunk
CHUNK_SIZE	NUMBER	Amount of memory allocated for this chunk
ALLOC_CLASS	VARCHAR2(8)	Class of memory that this chunk of memory belongs to. It will usually be either FREEABLE or PERMANENT.
CHUNK_TYPE	NUMBER	An index into a table of callback functions that tell the server how to re-create this chunk of memory should it need to be removed from the shared pool based on an LRU algorithm
SUBHEAP_DESC	RAW(4 8)	If the parent heap of this context heap is itself a subheap, then this is the address of the descriptor of the parent heap

V\$SQL_WORKAREA

V\$SQL_WORKAREA displays information about work areas used by SQL cursors. Each SQL statement stored in the shared pool has one or more child cursors that are listed in the V\$SQL view. V\$SQL_WORKAREA lists all work areas needed by these child cursors; V\$SQL_WORKAREA can be joined with V\$SQLAREA on (ADDRESS, HASH_VALUE) and with V\$SQL on (ADDRESS, HASH_VALUE, CHILD_NUMBER).

You can use this view to find out answers to the following questions:

- What are the top 10 work areas that require the most cache area?
- For work areas allocated in AUTO mode, what percentage of work areas are running using maximum memory?

Column	Datatype	Description
ADDRESS	RAW(4 8)	Address of the parent cursor handle
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. Two columns PARENT_HANDLE and HASH_VALUE can be used to join with V\$SQLAREA to locate the parent cursor.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache

Column	Datatype	Description
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns PARENT_HANDLE, HASH_VALUE, and CHILD_NUMBER can be used to join with V\$SQL to locate the child cursor using this area.
WORKAREA_ADDRESS	RAW(4 8)	Address of the work area handle. This is the primary key for the view.
OPERATION_TYPE	VARCHAR2(20)	Type of operation using the work area (SORT, HASH JOIN, GROUP BY, BUFFERING, BITMAP MERGE, or BITMAP CREATE)
OPERATION_ID	NUMBER	A unique number used to identify the operation in the execution plan. This identifier can be joined to V\$SQL_PLAN to locate the operation that uses this work area.
POLICY	VARCHAR2(10)	Sizing policy for this work area (MANUAL or AUTO)
ESTIMATED_OPTIMAL_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation completely in memory (optimal execution). Derived from either optimizer statistics or previous executions.
ESTIMATED_ONEPASS_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation in a single pass. Derived from either optimizer statistics or previous executions.
LAST_MEMORY_USED	NUMBER	Memory (in KB) used by this work area during the last execution of the cursor
LAST_EXECUTION	VARCHAR2(10)	Indicates whether this work area runs using OPTIMAL, ONE PASS, or ONE PASS memory requirement (or MULTI-PASS), during the last execution of the cursor
LAST_DEGREE	NUMBER	Degree of parallelism used during the last execution of this operation
TOTAL_EXECUTIONS	NUMBER	Number of times this work area was active
OPTIMAL_EXECUTIONS	NUMBER	Number of times this work area ran in optimal mode
ONEPASS_EXECUTIONS	NUMBER	Number of times this work area ran in one-pass mode
MULTIPASSES_EXECUTIONS	NUMBER	Number of times this work area ran below the one-pass memory requirement
ACTIVE_TIME	NUMBER	Average time this work area is active (in hundredths of a second)
MAX_TEMPSEG_SIZE	NUMBER	Maximum temporary segment size (in bytes) created by an instantiation of this work area. This column is null if this work area has never spilled to disk.
LAST_TEMPSEG_SIZE	NUMBER	Temporary segment size (in bytes) created in the last instantiation of this work area. This column is null if the last instantiation of this work area did not spill to disk.

V\$SQL_WORKAREA_ACTIVE

V\$SQL_WORKAREA_ACTIVE contains an instantaneous view of the work areas currently allocated by the system. You can join this view against V\$SQL_WORKAREA on WORKAREA_ADDRESS to access the definition of that work area. If a work area spills to disk, then this view contains information for the temporary segment created on behalf of this work area.

The last three columns are included to enable joining V\$SQL_WORKAREA_ACTIVE with V\$TEMPSEG_USAGE to retrieve more information on this temporary segment.

You can use this view to answer the following:

- What are the top 10 largest work areas currently allocated in my system?
- What percentage of memory is over-allocated (EXPECTED_SIZE < ACTUAL_MEM_USED) and under-allocated (EXPECTED_SIZE > ACTUAL_MEM_USED)?
- What are the active work areas using more memory than what is expected by the memory manager?
- What are the active work areas that have spilled to disk?

Column	Datatype	Description
SQL_HASH_VALUE	NUMBER	Hash value of the SQL statement that is currently being executed
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement that is currently being executed
WORKAREA_ADDRESS	RAW(4 8)	Address of the work area handle. This is the primary key for the view.
OPERATION_TYPE	VARCHAR2(20)	Type of operation using the work area (SORT, HASH JOIN, GROUP BY, BUFFERING, BITMAP MERGE, or BITMAP CREATE)
OPERATION_ID	NUMBER	A unique number used to identify the operation in the execution plan. This identifier can be joined to V\$SQL_PLAN to locate the operation that uses this work area.
POLICY	VARCHAR2(6)	Sizing policy for this work area (MANUAL or AUTO)
SID	NUMBER	Session identifier
QCINST_ID	NUMBER	Query coordinator instance identifier. Along with QCSID, enables you to uniquely identify the query coordinator.
QCSID	NUMBER	Query coordinator session identifier. This is the same as the SID if the work area is allocated by a serial cursor.
ACTIVE_TIME	NUMBER	Average time this work area is active (in centi-seconds)
WORK_AREA_SIZE	NUMBER	Maximum size of the work area as it is currently used by the operation
EXPECTED_SIZE	NUMBER	Expected size (in KB) for this work area. EXPECTED_SIZE is set on behalf of the operation by the memory manager. Memory can be over-allocated when WORK_AREA_SIZE has a higher value than EXPECTED_SIZE. This can occur when the operation using this work area takes a long time to resize it.
ACTUAL_MEM_USED	NUMBER	Amount of PGA memory (in KB) currently allocated on behalf of this work area. This value should range between 0 and WORK_AREA_SIZE.
MAX_MEM_USED	NUMBER	Maximum memory amount (in KB) used by this work area
NUMBER_PASSES	NUMBER	Number of passes corresponding to this work area (0 if running in OPTIMAL mode)
TEMPSEG_SIZE	NUMBER	Size (in bytes) of the temporary segment used on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
TABLESPACE	VARCHAR2(31)	Tablespace name for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
SEGRFNO#	NUMBER	Relative file number within the tablespace for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
SEGBLK#	NUMBER	Block number for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.

See Also: *Oracle Database Performance Tuning Guide* for more information on how to monitor SQL work areas

V\$SQL_WORKAREA_HISTOGRAM

V\$SQL_WORKAREA_HISTOGRAM displays the cumulative work area execution statistics (cumulated since instance startup) for different work area groups. The work areas are split into 33 groups based on their optimal memory requirements with the requirements increasing in powers of two. That is, work areas whose optimal requirement varies from 0 KB to 1 KB, 1 KB to 2 KB, 2 KB to 4 KB, ... and 2 TB to 4 TB.

For each work area group, the V\$SQL_WORKAREA_HISTOGRAM view shows how many work areas in that group were able to run in optimal mode, how many were able to run in one-pass mode, and finally how many ran in multi-pass mode. The DBA can take a snapshot at the beginning and the end of a desired time interval to derive the same statistics for that interval.

Column	Datatype	Description
LOW_OPTIMAL_SIZE	NUMBER	Lower bound for the optimal memory requirement of work areas included in this row (bytes)
HIGH_OPTIMAL_SIZE	NUMBER	Upper bound for the optimal memory requirement of work areas included in this row (bytes)
OPTIMAL_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in optimal mode since instance startup
ONEPASS_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in one-pass mode since instance startup
MULTIPASSES_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in multi-pass mode since instance startup
TOTAL_EXECUTIONS	NUMBER	Sum of OPTIMAL_EXECUTIONS, ONEPASS_EXECUTIONS, and MULTIPASSES_EXECUTIONS

See Also: *Oracle Database Performance Tuning Guide* for detailed information on how to monitor automatic PGA memory performance using this view

V\$SQLAREA

V\$SQLAREA lists statistics on shared SQL area and contains one row per SQL string. It provides statistics on SQL statements that are in memory, parsed, and ready for execution.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
SHARABLE_MEM	NUMBER	Amount of shared memory used by a cursor. If multiple child cursors exist, then the sum of all shared memory used by all child cursors.
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of an open cursor. If multiple child cursors exist, the fixed sum of memory used for the lifetime of all the child cursors.
RUNTIME_MEM	NUMBER	Fixed amount of memory required during execution of a cursor. If multiple child cursors exist, the fixed sum of all memory required during execution of all the child cursors.
SORTS	NUMBER	Sum of the number of sorts that were done for all the child cursors
VERSION_COUNT	NUMBER	Number of child cursors that are present in the cache under this parent
LOADED_VERSIONS	NUMBER	Number of child cursors that are present in the cache and have their context heap (KGL heap 6) loaded
OPEN_VERSIONS	NUMBER	The number of child cursors that are currently open under this current parent
USERS_OPENING	NUMBER	Number of users that have any of the child cursors open
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Total number of executions, totalled over all the child cursors
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.

Column	Datatype	Description
USERS_EXECUTING	NUMBER	Total number of users executing the statement over all child cursors
LOADS	NUMBER	Number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	Timestamp of the parent creation time
INVALIDATIONS	NUMBER	Total number of invalidations over all the child cursors
PARSE_CALLS	NUMBER	Sum of all parse calls to all the child cursors under this parent
DISK_READS	NUMBER	Sum of the number of disk reads over all child cursors
DIRECT_WRITES	NUMBER	Sum of the number of direct writes over all child cursors
BUFFER_GETS	NUMBER	Sum of buffer gets over all child cursors
APPLICATION_WAIT_TIME	NUMBER	Application wait time
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time
USER_IO_WAIT_TIME	NUMBER	User I/O Wait Time
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time
JAVA_EXEC_TIME	NUMBER	Java execution time
ROWS_PROCESSED	NUMBER	Total number of rows processed on behalf of this SQL statement
COMMAND_TYPE	NUMBER	Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(25)	Mode under which the SQL statement was executed
PARSING_USER_ID	NUMBER	User ID of the user that has parsed the very first cursor under this parent
PARSING_SCHEMA_ID	NUMBER	Schema ID that was used to parse this child cursor
KEPT_VERSIONS	NUMBER	Number of child cursors that have been marked to be kept using the DBMS_SHARED_POOL package
ADDRESS	RAW(4 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
OLD_HASH_VALUE	NUMBER	Old SQL hash value
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	Hash value of the module that is named in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	Hash value of the action that is named in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction fails to serialize, producing ORA-08177 errors, totalled over all the child cursors
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
IS_OBSOLETE	VARCHAR2(1)	Indicates whether the cursor has become obsolete (Y) or not (N). This can happen if the number of child cursors is too large.
CHILD_LATCH	NUMBER	Child latch number that is protecting the cursor
PROGRAM_ID	NUMBER	Program identifier

V\$SQLTEXT

This view contains the text of SQL statements belonging to shared SQL cursors in the SGA.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Used with HASH_VALUE to uniquely identify a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify a cached cursor
SQL_ID	VARCHAR2(13)	SQL identifier of a cached cursor
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2(64)	A column containing one piece of the SQL text

V\$SQLTEXT_WITH_NEWLINES

This view is identical to the V\$SQLTEXT view except that, to improve legibility, V\$SQLTEXT_WITH_NEWLINES does not replace newlines and tabs in the SQL statement with spaces.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Used with HASH_VALUE to uniquely identify a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify a cached cursor
SQL_ID	VARCHAR2(13)	SQL identifier of a cached cursor
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2(64)	A column containing one piece of the SQL text

See Also: ["V\\$SQLTEXT"](#) on page 5-69

V\$STANDBY_LOG

V\$STANDBY_LOG displays information about standby redo logs. Standby redo logs are similar to online redo logs, but standby redo logs are only used on a standby database that is receiving redo data from the primary database.

Column	Datatype	Description
GROUP#	NUMBER	Log group number
DBID	VARCHAR2(40)	Database ID of the primary database to which the standby redo logfile is assigned. If the standby redo logfile is unassigned, the value UNASSIGNED will be displayed.
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
BYTES	NUMBER	Size of the log (in bytes)
USED	NUMBER	Number of bytes used in the log
ARCHIVED	VARCHAR2(3)	Archive status (YES) or (NO)
STATUS	VARCHAR2(10)	Log status: <ul style="list-style-type: none"> ■ UNASSIGNED - Online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS when it is not the current redo log. ■ ACTIVE - Log is active but is not the current log. It is needed for failure recovery. It may be in use for block recovery. It may or may not be archived.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log

Column	Datatype	Description
LAST_CHANGE#	NUMBER	Last change number made to this datafile. Set to NULL if the datafile is being changed.
LAST_TIME	DATE	Time stamp of the last change

V\$STATISTICS_LEVEL

V\$STATISTICS_LEVEL displays the status of the statistics/advisories controlled by STATISTICS_LEVEL.

Column	Datatype	Description
STATISTICS_NAME	VARCHAR2(64)	Name of the statistic/advisory
DESCRIPTION	VARCHAR2(4000)	Description of the statistic/advisory
SESSION_STATUS	VARCHAR2(8)	Status of the statistic/advisory for the session: <ul style="list-style-type: none"> ▪ ENABLED ▪ DISABLED
SYSTEM_STATUS	VARCHAR2(8)	System-wide status of the statistic/advisory: <ul style="list-style-type: none"> ▪ ENABLED ▪ DISABLED
ACTIVATION_LEVEL	VARCHAR2(7)	Indicates the level of STATISTICS_LEVEL that enables the statistic/advisory: <ul style="list-style-type: none"> ▪ BASIC ▪ TYPICAL ▪ ALL
STATISTICS_VIEW_NAME	VARCHAR2(64)	If there is a single view externalizing the statistic/advisory, then this column contains the name of that view. If there is no such view, then this column is null. If there are multiple views involved, then the DESCRIPTION column mentions the view names.
SESSION_SETTABLE	VARCHAR2(3)	Indicates whether the statistic/advisory can be set at the session level (YES) or not (NO)

V\$STATNAME

This view displays decoded statistic names for the statistics shown in the V\$SESSTAT and V\$SYSSTAT tables.

On some platforms, the NAME and CLASS columns contain additional operating system-specific statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number Note: Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2(64)	Statistic name

Column	Datatype	Description
CLASS	NUMBER	A number representing one or more statistics classes. The following class numbers are additive: <ul style="list-style-type: none"> ▪ 1 - User ▪ 2 - Redo ▪ 4 - Enqueue ▪ 8 - Cache ▪ 16 - OS ▪ 32 - Real Application Clusters ▪ 64 - SQL ▪ 128 - Debug
STAT_ID	NUMBER	Identifier of the statistic

See Also:

- ["V\\$SESSTAT"](#) on page 5-47 and ["V\\$SYSSTAT"](#) on page 5-80
- [Appendix E, "Statistics Descriptions"](#) for a description of each statistic
- Your operating system-specific Oracle documentation

V\$STREAMS_APPLY_COORDINATOR

V\$STREAMS_APPLY_COORDINATOR displays information about each apply process coordinator. The coordinator for an apply process gets transactions from the apply process reader and passes them to apply servers.

Column	Datatype	Description
SID	NUMBER	Session ID of the coordinator's session
SERIAL#	NUMBER	Serial number of the coordinator's session
STATE	VARCHAR2(21)	State of the coordinator: <ul style="list-style-type: none"> ▪ INITIALIZING ▪ APPLYING ▪ SHUTTING DOWN CLEANLY ▪ ABORTING
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process
TOTAL_APPLIED	NUMBER	Total number of transactions applied by the apply process since the apply process was last started
TOTAL_WAIT_DEPS	NUMBER	Number of times since the apply process was last started that an apply server waited to apply a logical change record (LCR) in a transaction until another apply server applied a transaction because of a dependency between the transactions
TOTAL_WAIT_COMMITS	NUMBER	Number of times since the apply process was last started that an apply server waited to commit a transaction until another apply server committed a transaction to serialize commits
TOTAL_ADMIN	NUMBER	Number of administrative jobs issued since the apply process was last started
TOTAL_ASSIGNED	NUMBER	Number of transactions assigned to apply servers since the apply process was last started
TOTAL_RECEIVED	NUMBER	Total number of transactions received by the coordinator process since the apply process was last started

Column	Datatype	Description
TOTAL_IGNORED	NUMBER	Number of transactions which were received by the coordinator but were ignored because they had been previously applied
TOTAL_ROLLBACKS	NUMBER	Number of transactions which were rolled back due to unexpected contention
TOTAL_ERRORS	NUMBER	Number of transactions applied by the apply process that resulted in an apply error since the apply process was last started
LWM_TIME	DATE	Time when the message with the lowest message number was recorded. The creation time of the message with the lowest message number was also recorded at this time.
LWM_MESSAGE_NUMBER	NUMBER	Number of the message corresponding to the low-watermark. That is, messages with a commit message number less than or equal to this message number have definitely been applied, but some messages with a higher commit message number also may have been applied.
LWM_MESSAGE_CREATE_TIME	DATE	For captured messages, creation time at the source database of the message corresponding to the low-watermark. For user-enqueued messages, time when the message corresponding to the low-watermark was enqueued into the queue at the local database.
HWM_TIME	DATE	Time when the message with the highest message number was recorded. The creation time of the message with the highest message number was also recorded at this time.
HWM_MESSAGE_NUMBER	NUMBER	Number of the message corresponding to the high-watermark. That is, no messages with a commit message number greater than this message number have been applied.
HWM_MESSAGE_CREATE_TIME	DATE	For captured messages, creation time at the source database of the message corresponding to the high-watermark. For user-enqueued messages, time when the message corresponding to the high-watermark was enqueued into the queue at the local database.
STARTUP_TIME	DATE	Time when the apply process was last started
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed (in hundredths of a second) scheduling messages since the apply process was last started

Note: The ELAPSED_SCHEDULE_TIME column is only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V\$STREAMS_APPLY_READER

V\$STREAMS_APPLY_READER displays information about each apply reader. The apply reader for an apply process is a process which reads (dequeues) messages from the queue, computes message dependencies, builds transactions, and passes the transactions on to the apply process coordinator in commit order for assignment to the apply servers.

Column	Datatype	Description
SID	NUMBER	Session ID of the reader's session
SERIAL#	NUMBER	Serial number of the reader's session
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process

Column	Datatype	Description
STATE	VARCHAR2(17)	State of the reader: <ul style="list-style-type: none"> ■ IDLE ■ DEQUEUE_MESSAGES ■ SCHEDULE_MESSAGES
TOTAL_MESSAGES_DEQUEUED	NUMBER	Total number of messages dequeued since the apply process was last started
DEQUEUE_TIME	DATE	Time when the last message was received
DEQUEUED_MESSAGE_NUMBER	NUMBER	Number of the last message received
DEQUEUED_MESSAGE_CREATE_TIME	DATE	For captured messages, creation time at the source database of the last message received. For user-enqueued messages, time when the message was enqueued into the queue at the local database.
SGA_USED	NUMBER	Amount (in bytes) of SGA memory used by the apply process since it was last started
ELAPSED_DEQUEUE_TIME	NUMBER	Time elapsed (in hundredths of a second) dequeuing messages since the apply process was last started
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed (in hundredths of a second) scheduling messages since the apply process was last started. Scheduling includes computing dependencies between messages and assembling messages into transactions.
LAST_BROWSE_NUM	NUMBER	Last browse SCN
OLDEST_SCN_NUM	NUMBER	Oldest SCN
LAST_BROWSE_SEQ	NUMBER	Last browse sequence number
LAST_DEQ_SEQ	NUMBER	Last dequeue sequence number

Note: The ELAPSED_DEQUEUE_TIME and ELAPSED_SCHEDULE_TIME columns are only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V\$STREAMS_APPLY_SERVER

V\$STREAMS_APPLY_SERVER displays information about each apply server and its activities. An apply server receives events from the apply coordinator for an apply process. For each event received, an apply server either applies the event or sends the event to the appropriate apply handler.

Column	Datatype	Description
SID	NUMBER	Session ID of the apply server's session
SERIAL#	NUMBER	Serial number of the apply server's session
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process
SERVER_ID	NUMBER	Parallel execution server number of the apply server

Column	Datatype	Description
STATE	VARCHAR2(20)	State of the apply server: <ul style="list-style-type: none"> ■ IDLE - Performing no work ■ RECORD LOW-WATERMARK - Performing an administrative job that maintains information about the apply progress, which is used in the ALL_APPLY_PROGRESS and DBA_APPLY_PROGRESS data dictionary views ■ ADD PARTITION - Performing an administrative job that adds a partition that is used for recording information about in-progress transactions ■ DROP PARTITION - Performing an administrative job that drops a partition that was used to record information about in-progress transactions ■ EXECUTE TRANSACTION - Applying a transaction ■ WAIT COMMIT - Waiting to commit a transaction until all other transactions with a lower commit SCN are applied. This state is possible only if the COMMIT_SERIALIZATION apply process parameter is set to a value other than none and the PARALLELISM apply process parameter is set to a value greater than 1. ■ WAIT DEPENDENCY - Waiting to apply a logical change record (LCR) in a transaction until another transaction, on which it has a dependency, is applied. This state is possible only if the PARALLELISM apply process parameter is set to a value greater than 1. ■ WAIT FOR NEXT CHUNK - Waiting for the next set of logical change records (LCRs) for a large transaction
XIDUSN	NUMBER	Transaction ID undo segment number of the transaction currently being applied
XIDSLT	NUMBER	Transaction ID slot number of the transaction currently being applied
XIDSQN	NUMBER	Transaction ID sequence number of the transaction currently being applied
COMMITSCN	NUMBER	Commit system change number (SCN) of the transaction currently being applied
DEP_XIDUSN	NUMBER	Transaction ID undo segment number of a transaction on which the transaction being applied by this apply server depends
DEP_XIDSLT	NUMBER	Transaction ID slot number of a transaction on which the transaction being applied by this apply server depends
DEP_XIDSQN	NUMBER	Transaction ID sequence number of a transaction on which the transaction being applied by this apply server depends
DEP_COMMITSCN	NUMBER	Commit system change number (SCN) of the transaction on which this apply server depends
MESSAGE_SEQUENCE	NUMBER	Number of the current message being applied by the apply server. This value is reset to 1 at the beginning of each transaction.
TOTAL_ASSIGNED	NUMBER	Total number of transactions assigned to the apply server since the apply process was last started
TOTAL_ADMIN	NUMBER	Total number of administrative jobs done by the apply server since the apply process was last started. See the STATE information in this view for the types of administrative jobs.
TOTAL_ROLLBACKS	NUMBER	Number of transactions assigned to this server which were rolled back
TOTAL_MESSAGES_APPLIED	NUMBER	Total number of messages applied by this apply server since the apply process was last started
APPLY_TIME	DATE	Time last message was applied
APPLIED_MESSAGE_NUMBER	NUMBER	Number of the last message applied
APPLIED_MESSAGE_CREATE_TIME	DATE	Creation time at the source database of the last captured message applied. No information about user-enqueued messages is recorded in this column.

V\$STREAMS_CAPTURE

Column	Datatype	Description
ELAPSED_DEQUEUE_TIME	NUMBER	Time elapsed (in hundredths of a second) dequeuing messages since the apply process was last started
ELAPSED_APPLY_TIME	NUMBER	Time elapsed (in hundredths of a second) applying messages since the apply process was last started

Note: The `ELAPSED_DEQUEUE_TIME` and `ELAPSED_APPLY_TIME` columns are only populated if the `TIMED_STATISTICS` initialization parameter is set to `true`, or if the `STATISTICS_LEVEL` initialization parameter is set to `TYPICAL` or `ALL`.

V\$STREAMS_CAPTURE

`V$STREAMS_CAPTURE` displays information about each capture process.

Column	Datatype	Description
SID	NUMBER	Session identifier of the capture process
SERIAL#	NUMBER	Session serial number of the capture process session
CAPTURE#	NUMBER	Capture process number. A capture process is an Oracle background process, prefixed by <code>cp</code> .
CAPTURE_NAME	VARCHAR2(30)	Name of the capture process
LOGMINER_ID	NUMBER	Session ID of the LogMiner session associated with the capture process
STARTUP_TIME	DATE	Time when the capture process was last started
STATE	VARCHAR2(161)	State of the capture process: <ul style="list-style-type: none">INITIALIZINGCAPTURING CHANGESEVALUATING RULEENQUEUEING MESSAGESHUTTING DOWNABORTINGCREATING LCRWAITING FOR DICTIONARY REDOWAITING FOR REDOPAUSED FOR FLOW CONTROLDICTIONARY INITIALIZATION
TOTAL_PREFILTER_DISCARDED	NUMBER	Total number of prefiltered messages discarded
TOTAL_PREFILTER_KEPT	NUMBER	Total number of prefiltered messages kept
TOTAL_PREFILTER_EVALUATIONS	NUMBER	Total number of prefilter evaluations
TOTAL_MESSAGES_CAPTURED	NUMBER	Total changes captured since the capture process was last started
CAPTURE_TIME	DATE	Time when the most recent message was captured
CAPTURE_MESSAGE_NUMBER	NUMBER	Number of the most recently captured message
CAPTURE_MESSAGE_CREATE_TIME	DATE	Creation time of the most recently captured message
Total_messages_created	NUMBER	Count associated with <code>ELAPSED_LCR_TIME</code> to calculate rate

Column	Datatype	Description
Total_full_evaluations	NUMBER	Count associated with ELAPSED_RULE_TIME to calculate rate
TOTAL_MESSAGES_ENQUEUED	NUMBER	Total number of messages enqueued since the capture process was last started
ENQUEUE_TIME	DATE	Time when the last message was enqueued
ENQUEUE_MESSAGE_NUMBER	NUMBER	Number of the last enqueued message
ENQUEUE_MESSAGE_CREATE_TIME	DATE	Creation time of the last enqueued message
Available_message_number	NUMBER	For local capture, the last redo SCN flushed to the logfiles. For downstream capture, the last SCN added to LogMiner via the archive logs.
AVAILABLE_MESSAGE_CREATE_TIME	DATE	For local capture, the time the SCN was written to the logfile. For downstream capture, the time the most recent archive log (containing the most recent SCN) was added to LogMiner.
ELAPSED_CAPTURE_TIME	NUMBER	Elapsed time (in hundredths of a second) scanning for changes in the redo log since the capture process was last started
ELAPSED_RULE_TIME	NUMBER	Elapsed time (in hundredths of a second) evaluating rules since the capture process was last started
ELAPSED_ENQUEUE_TIME	NUMBER	Elapsed time (in hundredths of a second) enqueueing messages since the capture process was last started
ELAPSED_LCR_TIME	NUMBER	Elapsed time (in hundredths of a second) creating logical change records (LCRs) since the capture process was last started
ELAPSED_REDO_WAIT_TIME	NUMBER	Elapsed time (in hundredths of a second) spent by the capture process in the WAITING FOR REDO state

Note: The ELAPSED_CAPTURE_TIME, ELAPSED_RULE_TIME, ELAPSED_ENQUEUE_TIME, ELAPSED_LCR_TIME, and ELAPSED_REDO_WAIT_TIME columns are only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V\$SUBCACHE

This view displays information about the subordinate caches currently loaded into library cache memory. The view walks through the library cache, printing out a row for each loaded subordinate cache per library cache object.

Column	Datatype	Description
OWNER_NAME	VARCHAR2(64)	Owner of object containing these cache entries
NAME	VARCHAR2(1000)	Object Name
TYPE	NUMBER	Object Type
HEAP_NUM	NUMBER	Heap number containing this subordinate cache
CACHE_ID	NUMBER	Subordinate cache ID
CACHE_CNT	NUMBER	Number of entries for this cache in this object
HEAP_SZ	NUMBER	Amount of extent space allocated to this heap
HEAP_ALOC	NUMBER	Amount of extent space allocated from this heap
HEAP_USED	NUMBER	Amount of space utilized in this heap

V\$SYS_OPTIMIZER_ENV

V\$SYS_OPTIMIZER_ENV displays the contents of the optimizer environment for the instance. The optimizer environment stores the value of the main parameters used by the Oracle optimizer when building the execution plan of a SQL statement. Hence, modifying the value of one or more of these parameters (for example, by issuing an ALTER SYSTEM statement) could lead to plan changes.

The parameters displayed by this view are either regular initialization parameters (such as OPTIMIZER_FEATURES_ENABLE) or pseudo parameters (such as ACTIVE_INSTANCE_COUNT).

Column	Datatype	Description
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2 (40)	Name of the parameter
ISDEFAULT	VARCHAR2 (3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2 (25)	Value of the parameter
DEFAULT_VALUE	VARCHAR2 (25)	Default value of the parameter

V\$SYS_TIME_MODEL

V\$SYS_TIME_MODEL displays the system-wide accumulated times for various operations. The time reported is the total elapsed or CPU time (in microseconds). Any timed operation will buffer at most 5 seconds of time data. Specifically, this means that if a timed operation (such as SQL execution) takes a long period of time to perform, the data published to this view is at most missing 5 seconds of the time accumulated for the operation.

The time values are 8-byte integers and can therefore hold approximately 580,000 years worth of time before wrapping. Background process time is not included in a statistic value unless the statistic is specifically for background processes.

Column	Datatype	Description
STAT_ID	NUMBER	Statistic identifier for the time statistic
STAT_NAME	VARCHAR2 (64)	Name of the statistic (see Table 5-4 on page 5-37)
VALUE	NUMBER	Amount of time (in microseconds) that the system has spent in this operation

V\$SYSAUX_OCCUPANTS

V\$SYSAUX_OCCUPANTS displays SYSAUX tablespace occupant information.

Column	Datatype	Description
OCCUPANT_NAME	VARCHAR2 (64)	Occupant name
OCCUPANT_DESC	VARCHAR2 (64)	Occupant description
SCHEMA_NAME	VARCHAR2 (64)	Schema name for the occupant
MOVE_PROCEDURE	VARCHAR2 (64)	Name of the move procedure; null if not applicable
MOVE_PROCEDURE_DESC	VARCHAR2 (64)	Description of the move procedure
SPACE_USAGE_KBYTES	NUMBER	Current space usage of the occupant (in KB)

V\$SYSMETRIC

V\$SYSMETRIC displays the system metric values captured for the most current time interval for both the long duration (60-second) and short duration (15-second) system metrics.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2 (64)	Metric name
VALUE	NUMBER	Metric value
METRIC_UNIT	VARCHAR2 (64)	Metric unit description

V\$SYSMETRIC_HISTORY

V\$SYSMETRIC_HISTORY displays all system metric values available in the database. Both long duration (60-second with 1 hour history) and short duration (15-second with one-interval only) metrics are displayed by this view.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2 (64)	Metric name
VALUE	NUMBER	Metric value
METRIC_UNIT	VARCHAR2 (64)	Metric unit description

V\$SYSMETRIC_SUMMARY

V\$SYSMETRIC_SUMMARY displays a summary of all system Metric values for the long-duration system metrics. The average, maximum value, minimum value, and the value of one standard deviation for the last hour are displayed for each metric item.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2 (64)	Metric name
NUM_INTERVAL	NUMBER	Number of intervals observed
MAXVAL	NUMBER	Maximum value observed
MINVAL	NUMBER	Minimum value observed

Column	Datatype	Description
AVERAGE	NUMBER	Average value over the period
STANDARD_DEVIATION	NUMBER	One standard deviation
METRIC_UNIT	VARCHAR2(64)	Metric unit description

V\$SYSSTAT

This view lists system statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V\$STATNAME view.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number Note: Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2(64)	Statistic name
CLASS	NUMBER	A number representing one or more statistics class. The following class numbers are additive: <ul style="list-style-type: none"> ▪ 1 - User ▪ 2 - Redo ▪ 4 - Enqueue ▪ 8 - Cache ▪ 16 - OS ▪ 32 - Real Application Clusters ▪ 64 - SQL ▪ 128 - Debug
VALUE	NUMBER	Statistic value
STAT_ID	NUMBER	Identifier of the statistic

See Also: ["V\\$STATNAME"](#) on page 5-71 and [Appendix E, "Statistics Descriptions"](#)

V\$SYSTEM_CURSOR_CACHE

This view displays similar information to the V\$SESSION_CURSOR_CACHE view except that this information is system wide.

See Also: ["V\\$SESSION_CURSOR_CACHE"](#) on page 5-42

Column	Datatype	Description
OPENS	NUMBER	Cumulative total of cursor opens
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times you found an open cursor divided by the number of times you looked for a cursor

V\$SYSTEM_EVENT

This view contains information on total waits for an event. Note that the TIME_WAITED and AVERAGE_WAIT columns will contain a value of zero on those platforms that do not support a fast timing mechanism. If you are running on one of these

platforms and you want this column to reflect true wait times, you must set `TIMED_STATISTICS` to `TRUE` in the parameter file; doing this will have a small negative effect on system performance.

See Also: ["TIMED_STATISTICS"](#) on page 1-139

Column	Datatype	Description
EVENT	VARCHAR2(64)	Name of the wait event
TOTAL_WAITS	NUMBER	Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event
TIME_WAITED	NUMBER	Total amount of time waited for the event (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event (in hundredths of a second)
TIME_WAITED_MICRO	NUMBER	Total amount of time waited for the event (in microseconds)
EVENT_ID	NUMBER	Identifier of the wait event

V\$SYSTEM_PARAMETER

`V$SYSTEM_PARAMETER` displays information about the initialization parameters that are currently in effect for the instance. A new session inherits parameter values from the instance-wide values.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter
TYPE	NUMBER	Parameter type: <ul style="list-style-type: none"> ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	Instance-wide parameter value
DISPLAY_VALUE	VARCHAR2(512)	Parameter value in a user-friendly format. For example, if the <code>VALUE</code> column shows the value 262144 for a big integer parameter, then the <code>DISPLAY_VALUE</code> column will show the value 256K.
ISDEFAULT	VARCHAR2(9)	Indicates whether the parameter is set to the default value (<code>TRUE</code>) or the parameter value was specified in the parameter file (<code>FALSE</code>)
ISSSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with <code>ALTER SESSION</code> (<code>TRUE</code>) or not (<code>FALSE</code>)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with <code>ALTER SYSTEM</code> and when the change takes effect: <ul style="list-style-type: none"> ■ <code>IMMEDIATE</code> - Parameter can be changed with <code>ALTER SYSTEM</code> regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ <code>DEFERRED</code> - Parameter can be changed with <code>ALTER SYSTEM</code> regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ <code>FALSE</code> - Parameter cannot be changed with <code>ALTER SYSTEM</code> unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.

Column	Datatype	Description
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEM, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SYSTEM was performed, the value will be MODIFIED.
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update
HASH	NUMBER	Hash value for the parameter name

V\$SYSTEM_PARAMETER2

V\$SYSTEM_PARAMETER2 displays information about the initialization parameters that are currently in effect for the instance, with each list parameter value appearing as a row in the view. A new session inherits parameter values from the instance-wide values.

Presenting the list parameter values in this format enables you to quickly determine the values for a list parameter. For example, if a parameter value is a , b, then the V\$SYSTEM_PARAMETER view does not tell you if the parameter has two values (both a and b) or one value (a , b). V\$SYSTEM_PARAMETER2 makes the distinction between the list parameter values clear.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter
TYPE	NUMBER	Parameter type: <ul style="list-style-type: none"> ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	Parameter value
DISPLAY_VALUE	VARCHAR2(512)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(6)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)

Column	Datatype	Description
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: <ul style="list-style-type: none"> IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEM, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SYSTEM was performed, the value will be MODIFIED.
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
ORDINAL	NUMBER	Position (ordinal number) of the parameter value. Useful only for parameters whose values are lists of strings.
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update

V\$SYSTEM_WAIT_CLASS

V\$SYSTEM_WAIT_CLASS displays the instance-wide time totals for each registered wait class.

Column	Datatype	Description
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
TOTAL_WAITS	NUMBER	Number of times waits of the class occurred
TIME_WAITED	NUMBER	Amount of time spent in the wait by all sessions in the instance

V\$TABLESPACE

This view displays tablespace information from the control file.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
NAME	VARCHAR2(30)	Tablespace name
INCLUDED_IN_DATABASE_BACKUP	VARCHAR2(3)	Indicates whether the tablespace is included in full database backups using the BACKUP DATABASE RMAN command (YES) or not (NO). NO only if the CONFIGURE EXCLUDE RMAN command was used for this tablespace.
BIGFILE	VARCHAR2(3)	Indicates whether the tablespace is a bigfile tablespace (YES) or not (NO)

V\$TEMP_CACHE_TRANSFER

Column	Datatype	Description
FLASHBACK_ON	VARCHAR2(3)	Indicates whether the tablespace participates in FLASHBACK DATABASE operations (YES) or not (NO)

V\$TEMP_CACHE_TRANSFER

V\$TEMP_CACHE_TRANSFER is deprecated.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the tempfile
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
RBR	NUMBER	Number of reuse blocks cross-instance calls; always 0
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse blocks cross-instance calls; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0

V\$TEMP_EXTENT_MAP

This view displays the status of each unit for all LOCALLY MANAGED temporary tablespaces.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of tablespace this unit belongs to
FILE_ID	NUMBER	Absolute file number
BLOCK_ID	NUMBER	Begin block number for this unit
BYTES	NUMBER	Bytes in extent
BLOCKS	NUMBER	Blocks in extent
OWNER	NUMBER	Which instance own this unit (string)
RELATIVE_FNO	NUMBER	The relative file number

V\$TEMP_EXTENT_POOL

This view displays the state of temporary space cached and used for a given instance. Note that loading of the temporary space cache is lazy, and that instances can be dormant. Use GV\$TEMP_EXTENT_POOL for information about all instances.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace
FILE_ID	NUMBER	Absolute file number
EXTENTS_CACHED	NUMBER	How many extents have been cached

Column	Datatype	Description
EXTENTS_USED	NUMBER	How many extents are actually being used
BLOCKS_CACHED	NUMBER	How many blocks are cached
BLOCKS_USED	NUMBER	How many blocks are used
BYTES_CACHED	NUMBER	How many bytes are cached
BYTES_USED	NUMBER	How many bytes used
RELATIVE_FNO	NUMBER	The relative file number

V\$TEMP_HISTOGRAM

V\$TEMP_HISTOGRAM displays a histogram of all single block reads on a per-tempfile basis. The histogram has buckets of time intervals from < 1 ms, < 2 ms, < 4 ms, < 8 ms, ... < 2²¹ ms, < 2²² ms, >= 2²² ms.

The histogram will not be filled unless the TIMED_STATISTICS initialization parameter is set to true.

Column	Datatype	Description
FILE#	NUMBER	File number
SINGLEBLKRDTIM_MILLI	NUMBER	Amount of time the bucket represents (in milliseconds). If the duration = <i>num</i> , then this column represents waits of duration < <i>num</i> that are not included in any smaller bucket.
SINGLEBLKRDS	NUMBER	Number of waits of the duration belonging to the bucket of the histogram

V\$TEMP_SPACE_HEADER

This view displays aggregate information per file per LOCALLY MANAGED temporary tablespace regarding how much space is currently being used and how much is free as identified in the space header.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the temporary tablespace
FILE_ID	NUMBER	Absolute file number
BYTES_USED	NUMBER	How many bytes are in use
BLOCKS_USED	NUMBER	How many blocks are in use
BYTES_FREE	NUMBER	How many bytes are free
BLOCKS_FREE	NUMBER	How many blocks are free
RELATIVE_FNO	NUMBER	The relative file number for the file

V\$TEMPFILE

This view displays tempfile information.

Column	Datatype	Description
FILE#	NUMBER	Absolute file number
CREATION_CHANGE#	NUMBER	Creation System Change Number (SCN)
CREATION_TIME	DATE	Creation time
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Relative file number in the tablespace

Column	Datatype	Description
STATUS	VARCHAR2(7)	Status of the file (OFFLINE ONLINE)
ENABLED	VARCHAR2(10)	Enabled for read and/or write
BYTES	NUMBER	Size of the file in bytes (from the file header)
BLOCKS	NUMBER	Size of the file in blocks (from the file header)
CREATE_BYTES	NUMBER	Creation size of the file (in bytes)
BLOCK_SIZE	NUMBER	Block size for the file
NAME	VARCHAR2(513)	Name of the file

V\$TEMPORARY_LOBS

V\$TEMPORARY_LOBS displays temporary LOBs.

Column	Datatype	Description
SID	NUMBER	Session ID
CACHE_LOBS	NUMBER	Number of cache temp LOBs
NOCACHE_LOBS	NUMBER	Number of nocache temp LOBs
ABSTRACT_LOBS	NUMBER	Number of abstract LOBs

V\$TEMPSEG_USAGE

This view describes temporary segment usage.

Column	Datatype	Description
USERNAME	VARCHAR2(30)	User who requested temporary space
USER	VARCHAR2(30)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in USERNAME.
SESSION_ADDR	RAW(4 8)	Address of shared SQL cursor
SESSION_NUM	NUMBER	Serial number of session
SQLADDR	RAW(4 8)	Address of SQL statement
SQLHASH	NUMBER	Hash value of SQL statement
SQL_ID	VARCHAR2(13)	SQL identifier of SQL statement
TABLESPACE	VARCHAR2(31)	Tablespace in which space is allocated
CONTENTS	VARCHAR2(9)	Indicates whether tablespace is TEMPORARY or PERMANENT
SEGTYPE	VARCHAR2(9)	Type of sort segment: <ul style="list-style-type: none"> ▪ SORT ▪ HASH ▪ DATA ▪ INDEX ▪ LOB_DATA ▪ LOB_INDEX
SEGFILE#	NUMBER	File number of initial extent
SEGBLK#	NUMBER	Block number of the initial extent
EXTENTS	NUMBER	Extents allocated to the sort
BLOCKS	NUMBER	Extents in blocks allocated to the sort
SEGRFNO#	NUMBER	Relative file number of initial extent

V\$TEMPSTAT

This view contains information about file read/write statistics.

Column	Datatype	Description
FILE#	NUMBER	Number of the file
PHYRDS	NUMBER	Number of physical reads done
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
SINGLEBLKRDS	NUMBER	Number of single block reads
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER	Cumulative single block read time (in hundredths of a second)
AVGIOTIM	NUMBER	Average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is true; 0 if false
LSTIOTIM	NUMBER	Time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MINIOTIM	NUMBER	Minimum time (in hundredths of a second) spent on a single I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIORTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is true; 0 if false

V\$THREAD

This view contains thread information from the control file.

Column	Datatype	Description
THREAD#	NUMBER	Thread number
STATUS	VARCHAR2(6)	Thread status (OPEN CLOSED)
ENABLED	VARCHAR2(8)	Enabled status: DISABLED, (enabled) PRIVATE, or (enabled) PUBLIC
GROUPS	NUMBER	Number of log groups assigned to this thread
INSTANCE	VARCHAR2(16)	Instance name, if available
OPEN_TIME	DATE	Last time the thread was opened
CURRENT_GROUP#	NUMBER	Current log group
SEQUENCE#	NUMBER	Sequence number of current log
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Time of last checkpoint
ENABLE_CHANGE#	NUMBER	SCN at which thread was enabled
ENABLE_TIME	DATE	Time of enable SCN
DISABLE_CHANGE#	NUMBER	SCN at which thread was disabled
DISABLE_TIME	DATE	Time of disable SCN

V\$THRESHOLD_TYPES

V\$THRESHOLD_TYPES displays information about threshold types.

Column	Datatype	Description
METRICS_ID	NUMBER	Metrics ID
METRICS_GROUP_ID	NUMBER	Metrics group ID
OPERATOR_MASK	NUMBER	Operator mask
OBJECT_TYPE	VARCHAR2(64)	Object type: <ul style="list-style-type: none"> ▪ SYSTEM ▪ FILE ▪ SERVICE ▪ EVENT_CLASS ▪ TABLESPACE ▪ SESSION
ALERT_REASON_ID	NUMBER	ID of the alert reason

V\$TIMER

This view lists the elapsed time in hundredths of seconds. Time is measured since the beginning of the epoch, which is operating system specific, and wraps around to 0 again whenever the value overflows four bytes (roughly 497 days).

Column	Datatype	Description
HSECS	NUMBER	Elapsed time in hundredths of a second

V\$TIMEZONE_NAMES

V\$TIMEZONE_NAMES displays valid time zone names.

Column	Datatype	Description
TZNAME	VARCHAR2(64)	Time zone region (for example, US/Pacific)
TZABBREV	VARCHAR2(64)	Corresponding daylight abbreviation (for example, PDT)

V\$TRANSACTION

V\$TRANSACTION lists the active transactions in the system.

Column	Datatype	Description
ADDR	RAW(4 8)	Address of the transaction state object
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
UBAFIL	NUMBER	Undo block address (UBA) filenum
UBABLK	NUMBER	UBA block number
UBASQN	NUMBER	UBA sequence number
UBAREC	NUMBER	UBA record number
STATUS	VARCHAR2(16)	Status
START_TIME	VARCHAR2(20)	Start time (wall clock)

Column	Datatype	Description
START_SCNB	NUMBER	Start system change number (SCN) base
START_SCNW	NUMBER	Start SCN wrap
START_UEXT	NUMBER	Start extent number
START_UBAFIL	NUMBER	Start UBA file number
START_UBABLK	NUMBER	Start UBA block number
START_UBASQN	NUMBER	Start UBA sequence number
START_UBAREC	NUMBER	Start UBA record number
SES_ADDR	RAW (4 8)	User session object address
FLAG	NUMBER	Flag
SPACE	VARCHAR2 (3)	YES if a space transaction
RECURSIVE	VARCHAR2 (3)	YES if a recursive transaction
NOUNDO	VARCHAR2 (3)	YES if a no undo transaction
PTX	VARCHAR 2 (3)	YES if parallel transaction
NAME	VARCHAR2 (256)	Name of a named transaction
PRV_XIDUSN	NUMBER	Previous transaction undo segment number
PRV_XIDSLT	NUMBER	Previous transaction slot number
PRV_XIDSQN	NUMBER	Previous transaction sequence number
PTX_XIDUSN	NUMBER	Rollback segment number of the parent XID
PTX_XIDSLT	NUMBER	Slot number of the parent XID
PTX_XIDSQN	NUMBER	Sequence number of the parent XID
DSCN-B	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in DSCN_BASE.
DSCN-W	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in DSCN_WRAP.
USED_UBLK	NUMBER	Number of undo blocks used
USED_UREC	NUMBER	Number of undo records used
LOG_IO	NUMBER	Logical I/O
PHY_IO	NUMBER	Physical I/O
CR_GET	NUMBER	Consistent gets
CR_CHANGE	NUMBER	Consistent changes
START_DATE	DATE	Start time (wall clock)
DSCN_BASE	NUMBER	Dependent SCN base
DSCN_WRAP	NUMBER	Dependent SCN wrap
START_SCN	NUMBER	Start SCN
DEPENDENT_SCN	NUMBER	Dependent SCN
XID	RAW (8)	Transaction XID
PRV_XID	RAW (8)	Previous transaction XID
PTX_XID	RAW (8)	Parent transaction XID

V\$TRANSACTION_ENQUEUE

V\$TRANSACTION_ENQUEUE displays locks owned by transaction state objects.

V\$TRANSPORTABLE_PLATFORM

Column	Datatype	Description
ADDR	RAW(4 8)	Address of lock state object
KADDR	RAW(4 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. TX indicates transaction enqueue.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: <ul style="list-style-type: none">0 - none1 - null (NULL)2 - row-S (SS)3 - row-X (SX)4 - share (S)5 - S/Row-X (SSX)6 - exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: <ul style="list-style-type: none">0 - none1 - null (NULL)2 - row-S (SS)3 - row-X (SX)4 - share (S)5 - S/Row-X (SSX)6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V\$TRANSPORTABLE_PLATFORM

V\$TRANSPORTABLE_PLATFORM displays all platform names and their identifiers.

Column	Datatype	Description
PLATFORM_ID	NUMBER	Platform identification number
PLATFORM_NAME	VARCHAR2(101)	Platform name
ENDIAN_FORMAT	VARCHAR2(14)	Platform endian format: <ul style="list-style-type: none">BigLittleUNKNOWN FORMAT

V\$TYPE_SIZE

V\$TYPE_SIZE displays the sizes of various database components for use in estimating data block capacity.

Column	Datatype	Description
COMPONENT	VARCHAR2(8)	Component name, such as segment or buffer header
TYPE	VARCHAR2(8)	Component type
DESCRIPTION	VARCHAR2(32)	Description of the component
TYPE_SIZE	NUMBER	Size of the component

V\$UNDOSTAT

V\$UNDOSTAT displays a histogram of statistical data to show how well the system is working. The available statistics include undo space consumption, transaction concurrency, and length of queries executed in the instance. You can use this view to estimate the amount of undo space required for the current workload. Oracle uses this view to tune undo usage in the system. The view returns null values if the system is in manual undo management mode.

Each row in the view keeps statistics collected in the instance for a 10-minute interval. The rows are in descending order by the `BEGIN_TIME` column value. Each row belongs to the time interval marked by (`BEGIN_TIME`, `END_TIME`). Each column represents the data collected for the particular statistic in that time interval. The first row of the view contains statistics for the (partial) current time period. The view contains a total of 1008 rows, spanning a 7 day cycle.

Column	Datatype	Description
<code>BEGIN_TIME</code>	DATE	Identifies the beginning of the time interval
<code>END_TIME</code>	DATE	Identifies the end of the time interval
<code>UNDOTSN</code>	NUMBER	Represents the last active undo tablespace in the duration of time. The tablespace ID of the active undo tablespace is returned in this column. If more than one undo tablespace was active in that period, the active undo tablespace that was active at the end of the period is reported.
<code>UNDOBLKS</code>	NUMBER	Represents the total number of undo blocks consumed. You can use this column to obtain the consumption rate of undo blocks, and thereby estimate the size of the undo tablespace needed to handle the workload on your system.
<code>TXNCOUNT</code>	NUMBER	Identifies the total number of transactions executed within the period
<code>MAXQUERYLEN</code>	NUMBER	Identifies the length of the longest query (in seconds) executed in the instance during the period. You can use this statistic to estimate the proper setting of the <code>UNDO_RETENTION</code> initialization parameter. The length of a query is measured from the cursor open time to the last fetch/execute time of the cursor. Only the length of those cursors that have been fetched/executed during the period are reflected in the view.
<code>MAXQUERYID</code>	VARCHAR2(13)	SQL identifier of the longest running SQL statement in the period
<code>MAXCONCURRENCY</code>	NUMBER	Identifies the highest number of transactions executed concurrently within the period
<code>UNXPSTEALCNT</code>	NUMBER	Number of attempts to obtain undo space by stealing unexpired extents from other transactions
<code>UNXPBLKRELCNT</code>	NUMBER	Number of unexpired blocks removed from certain undo segments so they can be used by other transactions
<code>UNXPBLKREUCNT</code>	NUMBER	Number of unexpired undo blocks reused by transactions
<code>EXPSTEALCNT</code>	NUMBER	Number of attempts to steal expired undo blocks from other undo segments
<code>EXPBLKRELCNT</code>	NUMBER	Number of expired undo blocks stolen from other undo segments
<code>EXPBLKREUCNT</code>	NUMBER	Number of expired undo blocks reused within the same undo segments
<code>SSOLDERRCNT</code>	NUMBER	Identifies the number of times the error <code>ORA-01555</code> occurred. You can use this statistic to decide whether or not the <code>UNDO_RETENTION</code> initialization parameter is set properly given the size of the undo tablespace. Increasing the value of <code>UNDO_RETENTION</code> can reduce the occurrence of this error.
<code>NOSPACEERRCNT</code>	NUMBER	Identifies the number of times space was requested in the undo tablespace and there was no free space available. That is, all of the space in the undo tablespace was in use by active transactions. The corrective action is to add more space to the undo tablespace.
<code>ACTIVEBLKS</code>	NUMBER	Total number of blocks in the active extents of the undo tablespace for the instance at the sampled time in the period

V\$VERSION

Column	Datatype	Description
UNEXPIREDBLKs	NUMBER	Total number of blocks in the unexpired extents of the undo tablespace for the instance at the sampled time in the period
EXPIREDBLKs	NUMBER	Total number of blocks in the expired extents of the undo tablespace for the instance at the sampled time in the period
TUNED_UNDORETENTION	NUMBER	System tuned value indicating the period for which undo is being retained

V\$VERSION

V\$VERSION displays version numbers of core library components in the Oracle Database. There is one row for each component.

Column	Datatype	Description
BANNER	VARCHAR2(64)	Component name and version number

V\$VPD_POLICY

V\$VPD_POLICY displays all the fine-grained security policies and predicates associated with the cursors currently in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4 8)	Cursor address
PARADDR	RAW(4 8)	Parent cursor address
SQL_HASH	NUMBER	SQL hash number
SQL_ID	VARCHAR2(13)	SQL identifier
CHILD_NUMBER	NUMBER	Cursor's child number under the parent
OBJECT_OWNER	VARCHAR2(30)	Owner of the object with policy
OBJECT_NAME	VARCHAR2(30)	Name of the object with policy
POLICY_GROUP	VARCHAR2(30)	Name of the policy group
POLICY	VARCHAR2(30)	Name of the policy
POLICY_FUNCTION_OWNER	VARCHAR2(30)	Owner of the policy function
PREDICATE	VARCHAR2(4000)	Predicate for the policy (truncated to 4000 bytes in length)

V\$WAITCLASSMETRIC

V\$WAITCLASSMETRIC displays metric values of wait classes for the most recent 60-second interval. A history of the last one hour will be kept in the system.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
WAIT_CLASS#	NUMBER	Wait class number
WAIT_CLASS_ID	NUMBER	Wait class identifier
AVERAGE_WAITER_COUNT	NUMBER	Average waiter count
DBTIME_IN_WAIT	NUMBER	Database time spent in the wait
TIME_WAITED	NUMBER	Time waited during the interval
WAIT_COUNT	NUMBER	Number of times waited

V\$WAITCLASSMETRIC_HISTORY

V\$WAITCLASSMETRIC_HISTORY displays metric values of wait classes for all intervals in the last one hour.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
WAIT_CLASS#	NUMBER	Wait class number
WAIT_CLASS_ID	NUMBER	Wait class identifier
AVERAGE_WAITER_COUNT	NUMBER	Average waiter count
DBTIME_IN_WAIT	NUMBER	Database time spent in the wait
TIME_WAITED	NUMBER	Time waited during the interval
WAIT_COUNT	NUMBER	Number of times waited

V\$WAITSTAT

This view lists block contention statistics. This table is only updated when timed statistics are enabled.

Column	Datatype	Description
CLASS	VARCHAR2(18)	Class of the block
COUNT	NUMBER	Number of waits by this OPERATION for this CLASS of block
TIME	NUMBER	Sum of all wait times for all the waits by this OPERATION for this CLASS of block

Database Limits

This appendix lists the limits of values associated with database functions and objects. Limits exist on several levels in the database. There is usually a hard-coded limit in the database that cannot be exceeded. This value may be further restricted for any given operating system.

Database limits are divided into four categories:

- [Datatype Limits](#)
- [Physical Database Limits](#)
- [Logical Database Limits](#)
- [Process and Runtime Limits](#)

See Also: Your operating system-specific Oracle documentation

Datatype Limits

Datatypes	Limit	Comments
BFILE	Maximum size: 4 GB Maximum size of a file name: 255 characters Maximum size of a directory name: 30 characters Maximum number of open BFILEs: see Comments	The maximum number of BFILEs is limited by the value of the <code>SESSION_MAX_OPEN_FILES</code> initialization parameter, which is itself limited by the maximum number of open files the operating system will allow.
BLOB	Maximum size: (4 GB - 1) * DB_ BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000 ¹).
CHAR	Maximum size: 2000 bytes	
CHAR VARYING	Maximum size: 4000 bytes	
CLOB	Maximum size: (4 GB - 1) * DB_ BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000 ¹).
Literals (characters or numbers in SQL or PL/SQL)	Maximum size: 4000 characters	
LONG	Maximum size: 2 GB - 1	Only one LONG column is allowed per table.
NCHAR	Maximum size: 2000 bytes	

Datatypes	Limit	Comments
NCHAR VARYING	Maximum size: 4000 bytes	
NCLOB	Maximum size: (4 GB - 1) * DB_BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000 ¹).
NUMBER	999...(38 9's) x10 ¹²⁵ maximum value	Can be represented to full 38-digit precision (the mantissa).
	-999...(38 9's) x10 ¹²⁵ minimum value	Can be represented to full 38-digit precision (the mantissa).
Precision	38 significant digits	
RAW	Maximum size: 2000 bytes	
VARCHAR	Maximum size: 4000 bytes	
VARCHAR2	Maximum size: 4000 bytes	

¹ The absolute maximum number of columns in a table is 1000. However, when you create an object table (or a relational table with columns of object, nested table, varray, or REF type), Oracle maps the columns of the user-defined types to relational columns, creating in effect hidden columns that count toward the 1000-column limit. For details on how Oracle calculates the total number of columns in such a table, refer to *Oracle Database Administrator's Guide*.

See Also: *Oracle Database SQL Reference* for information on the rules and limitations governing the use of datatypes and naming database objects

Physical Database Limits

Item	Type of Limit	Limit Value
Database Block Size	Minimum	2048 bytes; must be a multiple of operating system physical block size
	Maximum	Operating system dependent; never more than 32 KB
Database Blocks	Minimum in initial extent of a segment.	2 blocks
	Maximum per datafile	Platform dependent; typically 2 ²² - 1 blocks
Controlfiles	Number of control files	1 minimum; 2 or more (on separate devices) strongly recommended
	Size of a control file	Dependent on operating system and database creation options; maximum of 20,000 x (database block size)
Database files	Maximum per tablespace	Operating system dependent; usually 1022
	Maximum per database	65533 May be less on some operating systems Limited also by size of database blocks and by the DB_FILES initialization parameter for a particular instance
Database extents	Maximum per dictionary managed tablespace	4 GB * physical block size (with K/M modifier); 4 GB (without K/M modifier)
	Maximum per locally managed (uniform) tablespace	2 GB * physical block size (with K/M modifier); 2 GB (without K/M modifier)
Database file size	Maximum	Operating system dependent. Limited by maximum operating system file size; typically 2 ²² or 4 MB blocks

Item	Type of Limit	Limit Value
MAXEXTENTS	Default value	Derived from tablespace default storage or DB_BLOCK_SIZE initialization parameter
	Maximum	Unlimited
Redo Log Files	Maximum number of logfiles	Limited by value of MAXLOGFILES parameter in the CREATE DATABASE statement Control file can be resized to allow more entries; ultimately an operating system limit
	Maximum number of logfiles per group	Unlimited
Redo Log File Size	Minimum size	50 KB
	Maximum size	Operating system limit; typically 2 GB
Tablespaces	Maximum number per database	64 KB Number of tablespaces cannot exceed the number of database files, as each tablespace must include at least one file
Bigfile Tablespaces	Number of blocks	2 ³² (4 GB) blocks
Smallfile (traditional) Tablespaces	Number of blocks	2 ²² (4 MB) blocks

Logical Database Limits

Item	Type	Limit
GROUP BY clause	Maximum length	The GROUP BY expression and all of the nondistinct aggregate functions (for example, SUM, AVG) must fit within a single database block.
Indexes	Maximum per table	Unlimited
	total size of indexed column	75% of the database block size minus some overhead
Columns	Per table	1000 columns maximum
	Per index (or clustered index)	32 columns maximum
	Per bitmapped index	30 columns maximum
Constraints	Maximum per column	Unlimited
Subqueries	Maximum levels of subqueries in a SQL statement	Unlimited in the FROM clause of the top-level query
		255 subqueries in the WHERE clause
Partitions	Maximum length of linear partitioning key	4 KB - overhead
	Maximum number of columns in partition key	16 columns
	Maximum number of partitions allowed per table or index	64 KB - 1 partitions
Rows	Maximum number per table	Unlimited

Item	Type	Limit
Stored Packages	Maximum size	PL/SQL and Developer/2000 may have limits on the size of stored procedures they can call. The limits typically range from 2000 to 3000 lines of code. See Also: Your PL/SQL or Developer/2000 documentation for details
Trigger Cascade Limit	Maximum value	Operating system-dependent, typically 32
Users and Roles	Maximum	2,147,483,638
Tables	Maximum per clustered table	32 tables
	Maximum per database	Unlimited

Process and Runtime Limits

Item	Type	Limit
Instances per database	Maximum number of cluster database instances per database	Operating system-dependent
Locks	Row-level	Unlimited
	Distributed Lock Manager	Operating system dependent
SGA size	Maximum value	Operating system-dependent; typically 2 to 4 GB for 32-bit operating systems, and > 4 GB for 64-bit operating systems
Advanced Queuing Processes	Maximum per instance	10
Job Queue Processes	Maximum per instance	1000
I/O Slave Processes	Maximum per background process (DBWR, LGWR, etc.)	15
	Maximum per Backup session	15
Sessions	Maximum per instance	32 KB; limited by the PROCESSES and SESSIONS initialization parameters
Global Cache Service Processes	Maximum per instance	10
Shared Servers	Maximum per instance	Unlimited within constraints set by the PROCESSES and SESSIONS initialization parameters, for instance
Dispatchers	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance
Parallel Execution Slaves	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance
Backup Sessions	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance

SQL Scripts

This appendix describes SQL scripts that are required for optimal operation of the Oracle Database.

The SQL scripts are described in the following sections:

- [Creating the Data Dictionary](#)
- [Creating Additional Data Dictionary Structures](#)
- [The "NO" Scripts](#)
- [Upgrade and Downgrade Scripts](#)
- [Java Scripts](#)

Note: Comments within the SQL scripts themselves contain more detailed information and examples.

Creating the Data Dictionary

When you use the Database Configuration Assistant to create a database, Oracle automatically creates the data dictionary. Thereafter, whenever the database is in operation, Oracle updates the data dictionary in response to every DDL statement.

The data dictionary base tables are the first objects created in any Oracle database. They are created in the `SYSTEM` tablespace and must remain there. The data dictionary base tables store information about all user-defined objects in the database.

[Table B-1](#) lists required scripts, which are run automatically when you create a database using the Database Configuration Assistant. They are described here because you might need to run them if you create a database manually, or if you upgrade and subsequently downgrade a database. To run these scripts, you must be connected to Oracle as a user with `SYSDBA` privileges.

Table B-1 *Creating the Data Dictionary Scripts*

Script Name	Needed For	Description
<code>catalog.sql</code>	All databases	Creates the data dictionary and public synonyms for many of its views Grants <code>PUBLIC</code> access to the synonyms
<code>catproc.sql</code>	All databases	Runs all scripts required for, or used with PL/SQL
<code>catclust.sql</code>	Real Application Clusters	Creates Real Application Clusters data dictionary views

See Also:

- Your operating system-specific Oracle documentation for the exact names and locations of these scripts on your operating system
- *Oracle Database Administrator's Guide* for more information about creating a database
- Your release notes and *Oracle Database Upgrade Guide* to learn when it is necessary to run these scripts

Creating Additional Data Dictionary Structures

Oracle supplies other scripts that create additional structures you can use in managing your database and creating database applications. These scripts are listed in [Table B-2](#).

See Also: Your operating system-specific Oracle documentation for the exact names and locations of these scripts on your operating system

Table B-2 *Creating Additional Data Dictionary Structures*

Script Name	Needed For	Run By	Description
catblock.sql	Performance management	SYS	Creates views that can dynamically display lock dependency graphs
catexp7.sql	Exporting data to Oracle7	SYS	Creates the dictionary views needed for the Oracle7 Export utility to export data from the Oracle Database in Oracle7 Export file format
caths.sql	Heterogeneous Services	SYS	Installs packages for administering heterogeneous services
catio.sql	Performance management	SYS	Allows I/O to be traced on a table-by-table basis
catoctk.sql	Security	SYS	Creates the Oracle Cryptographic Toolkit package
catqueue.sql	Advanced Queuing		Creates the dictionary objects required for Advanced Queuing
catrep.sql	Oracle Replication	SYS	Runs all SQL scripts for enabling database replication
catrman.sql	Recovery Manager	RMAN or any user with GRANT_RECOVERY_CATALOG_OWNERrole	Creates recovery manager tables and views (schema) to establish an external recovery catalog for the backup, restore, and recovery functionality provided by the Recovery Manager (RMAN) utility
dbmsiotc.sql	Storage management	Any user	Analyzes chained rows in index-organized tables
dbmsotrc.sql	Performance management	SYS or SYSDBA	Enables and disables generation of Oracle Trace output
dbmspool.sql	Performance management	SYS or SYSDBA	Enables DBA to lock PL/SQL packages, SQL statements, and triggers into the shared pool
userlock.sql	Concurrency control	SYS or SYSDBA	Provides a facility for user-named locks that can be used in a local or clustered environment to aid in sequencing application actions

Table B–2 (Cont.) Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
utlbstat.sql and utlestat.sql	Performance monitoring	SYS	Respectively start and stop collecting performance tuning statistics
utlchn1.sql	Storage management	Any user	For use with the Oracle Database. Creates tables for storing the output of the ANALYZE command with the CHAINED ROWS option. Can handle both physical and logical rowids.
utlconst.sql	Year 2000 compliance	Any user	Provides functions to validate that CHECK constraints on date columns are year 2000 compliant
utldtree.sql	Metadata management	Any user	Creates tables and views that show dependencies between objects
utlexpt1.sql	Constraints	Any user	For use with the Oracle Database. Creates the default table (EXCEPTIONS) for storing exceptions from enabling constraints. Can handle both physical and logical rowids.
utlip.sql	PL/SQL	SYS	Used primarily for upgrade and downgrade operations. It invalidates all existing PL/SQL modules by altering certain dictionary tables so that subsequent recompilations will occur in the format required by the database. It also reloads the packages STANDARD and DBMS_STANDARD, which are necessary for any PL/SQL compilations.
utlirp.sql	PL/SQL	SYS	Used to change from 32-bit to 64-bit word size or vice versa. This script recompiles existing PL/SQL modules in the format required by the new database. It first alters some data dictionary tables. Then it reloads the packages STANDARD and DBMS_STANDARD, which are necessary for using PL/SQL. Finally, it triggers a recompilation of all PL/SQL modules, such as packages, procedures, and types.
utllockt.sql	Performance monitoring	SYS or SYSDBA	Displays a lock wait-for graph, in tree structure format
utlpwdmg.sql	Security	SYS or SYSDBA	Creates PL/SQL functions for default password complexity verification. Sets the default password profile parameters and enables password management features.
utlrp.sql	PL/SQL	SYS	Recompiles all existing PL/SQL modules that were previously in an INVALID state, such as packages, procedures, and types.
utlsampl.sql	Examples	SYS or any user with DBA role	Creates sample tables, such as emp and dept, and users, such as scott
utlsc1n.sql	Oracle Replication	Any user	Copies a snapshot schema from another snapshot site

Table B–2 (Cont.) Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
utltkprf.sql	Performance management	SYS	Creates the TKPROFER role to allow the TKPROF profiling utility to be run by non-DBA users
utlvalid.sql	Partitioned tables	Any user	Creates tables required for storing output of ANALYZE TABLE ... VALIDATE STRUCTURE of a partitioned table
utlxplan.sql	Performance management	Any user	Creates the table PLAN_TABLE, which holds output from the EXPLAIN PLAN statement

The "NO" Scripts

The scripts in [Table B–3](#) are used to remove dictionary information for various optional services or components.

See Also: *Oracle Database Upgrade Guide* for more information about these scripts

Table B–3 The NO Scripts

Script Name	Needed For	Run By	Description
catnoadt.sql	Objects	SYS	Drops views and synonyms on dictionary metadata that relate to object types
catnoaud.sql	Security	SYS	Drops views and synonyms on auditing metadata
catnohs.sql	Heterogeneous Services	SYS	Removes Heterogeneous Services dictionary metadata
catnoprt.sql	Partitioning	SYS	Drops views and synonyms on dictionary metadata that relate to partitioned tables and indexes
catnoque.sql	Advanced Queuing	SYS	Removes Advanced Queuing dictionary metadata
catnormn.sql	Recovery Manager	Owner of recovery catalog	Removes recovery catalog schema
catnosvm.sql	Server Manager	SYS	Removes Oracle7 Server Manager views and synonyms
catnsnmp.sql	Distributed management	SYS	Drops the DBSNMP user and SNMPAGENT role

Upgrade and Downgrade Scripts

The scripts in [Table B–4](#) are used when upgrading or downgrading to another release of Oracle. To run these scripts, you must be connected to Oracle as a user with SYSDBA privileges.

Table B–4 Upgrade and Downgrade Scripts

Script Name	Needed For	Description
d0902000.sql	Downgrading	Provides a direct downgrade path from the current release of Oracle to release 9.2
u0800060.sql	Upgrading	Provides a direct upgrade path from release 8.0.6 of Oracle to the new release
u0801070.sql	Upgrading	Provides a direct upgrade path from release 8.1.7 of Oracle to the new release
u0900010.sql	Upgrading	Provides a direct upgrade path from release 9.0.1 of Oracle to the new release
u0902000.sql	Upgrading	Provides a direct upgrade path from release 9.2 of Oracle to the new release
utlu101i.SQL	Pre-Upgrade Information	Analyzes the database to be upgraded, detailing requirements and issues for the upgrade to release 10.1
utlu101s.SQL	Post-Upgrade Status	Displays the component upgrade status after an upgrade to release 10.1

See Also: *Oracle Database Upgrade Guide* for more information about these scripts

Java Scripts

The scripts in [Table B–5](#) are useful only if the JServer option is installed.

Table B–5 Java Scripts

Script Name	Description
initjvm.sql	Initializes JServer by installing core Java class libraries and Oracle-specific Java classes
rmjvm.sql	Removes all elements of the JServer
catjava.sql	Installs Java-related packages and classes

See Also: *Oracle Database Java Developer's Guide*

Oracle Wait Events

This appendix describes the event name, wait time, and parameters for wait events displayed by the `V$SESSION_WAIT` and `V$SYSTEM_EVENT` views.

This appendix contains the following topics:

- [Displaying Wait Events](#)
- [Wait Events and Parameters](#)
- [Parameter Descriptions](#)
- [Wait Event Descriptions](#)

Displaying Wait Events

Information about wait events is displayed in three dynamic performance views:

- `V$SESSION_WAIT` displays the events for which sessions have just completed waiting or are currently waiting.
- `V$SYSTEM_EVENT` displays the total number of times all the sessions have waited for the events in that view.
- `V$SESSION_EVENT` is similar to `V$SYSTEM_EVENT`, but displays all waits for each session.

See Also: "[V\\$SESSION_EVENT](#)" on page 5-43, "[V\\$SESSION_WAIT](#)" on page 5-45, and "[V\\$SYSTEM_EVENT](#)" on page 5-80

Many of these wait events are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice. Application developers should be aware of this and write their code to tolerate missing or extra wait events.

This appendix describes the event name, wait time, and parameters for each event.

The following SQL statement displays all Oracle wait events:

```
SQL> SELECT * FROM V$EVENT_NAME;
```

Wait Events and Parameters

The following wait events are present in the Oracle Database. The columns P1, P2, and P3 represent parameters for the wait event. Selected parameters and wait events are further described in the sections that follow.

Event Name	P1	P2	P3
alter system set dispatcher	waited		
ARCH random i/o			
ARCH sequential i/o			
ARCH wait for archivelog lock			
ARCH wait on ATTACH			
ARCH wait on DETACH			
ARCH wait on SENDREQ			
async disk IO	count	intr	timeout
block change tracking buffer space			
BFILE check if exists			
BFILE check if open			
BFILE closure			
BFILE get length			
BFILE get name object			
BFILE get path object			
BFILE internal seek			
BFILE open			
BFILE read			
buffer busy	group#	obj#	block#
buffer busy waits	file#	block#	class#
buffer deadlock	dba	class*10+mode	flag
buffer dirty disabled	group#		
buffer freelistbusy	group#	obj#	block#
buffer invalidation wait	group#	obj#	block#
buffer latch	latch addr	chain#	
buffer pool resize	buffer pool id	current size	new size
buffer read retry	file#	block#	
buffer rememberlist busy	group#	obj#	block#
buffer resize			
buffer write wait	group#	obj#	block#
buffer writeList full	group#	obj#	block#
CGS skgxn join retry	retry count		
CGS wait for IPC msg			
checkpoint completed			
contacting SCN server or SCN lock master			
control file heartbeat			
control file parallel write	files	blocks	requests
control file sequential read	file#	block#	blocks
control file single write	file#	block#	blocks
conversion file read	block#	blocks	
cr request retry	file#	block#	
db file parallel read	files	blocks	requests
db file parallel write	files	blocks	requests

Event Name	P1	P2	P3
db file scattered read	file#	block#	blocks
db file sequential read	file#	block#	blocks
db file single write	file#	block#	blocks
debugger command	not used	not used	not used
DFS db file lock	file#	not used	not used
DFS lock handle	type mode	id1	id2
direct path read	file number	first dba	block cnt
direct path read temp	file number	first dba	block cnt
direct path write	file number	first dba	block cnt
direct path write temp	file number	first dba	block cnt
dispatcher shutdown			
dispatcher timer	sleep time	not used	not used
DLM generic wait event			
dupl. cluster key	dba	not used	not used
enqueue	name mode	id1	id2
fib/fob latch			
file identify	fib	file name	opcode
file open	fib	iov	0
free buffer waits	file#	block#	set-id#
free global transaction table entry	tries		
free process state object			
gcs domain validation	cluinc	rcvinc	
gcs for action			
gcs remote message	waittime	poll	p3
gcs to be enabled			
ges cached resource cleanup	waittime		
ges cancel			
ges cgs de-registration			
ges cgs registration			
ges drm deferred close			
ges enter server mode			
ges generic event			
ges global resource directory to be frozen			
ges global resource directory to be unfrozen			
ges inquiry response	type mode	id1	id2
ges lmd and pmon to attach			
ges LMD suspend for testing event			
ges LMD to inherit communication channels			
ges LMD to shutdown			
ges lmd/lmses to freeze in rcfg - mrcvr			
ges lmd/lmses to freeze in rcfg - srcvr			
ges lmd/lmses to unfreeze in rcfg - mrcvr			
ges lmd/lmses to unfreeze in rcfg - srcvr			

Wait Events and Parameters

Event Name	P1	P2	P3
ges LMON for send buffers to be empty			
ges LMON for send queues			
ges LMON to get to FTDONE			
ges LMON to join CGS group			
ges LMS suspend for testing event			
ges master to get established for SCN op			
ges performance test completion			
ges pmon to exit			
ges proc table latch - kjdrptref			
ges process allocation			
ges process with outstanding i/o	pid		
ges reconfiguration to start			
ges remote message	waittime	loop	p3
ges resource cleanout during enqueue open			
ges resource cleanout during enqueue open-cvt			
ges reusing os pid	pid	count	
ges server qs to be empty - kjbrrit			
ges1 LMON to wake up LMD			
ges1 LMON to wake up LMD - mrcvr			
ges2 LMON to wake up LMD - mrcvr			
ges2 LMON to wake up LMD - srcvr			
ges2 LMON to wake up lms - mrcvr 2			
global cache assume wait	le		
global cache bg acks	count	loops	
global cache busy	file#	block#	id
global cache cancel wait	le		
global cache claim wait			
global cache cr cancel wait	le		
global cache cr disk request	file#	block#	le
global cache cr request	file#	block#	le
global cache domain validation			
global cache freelist wait			
global cache null to s	file#	block#	le
global cache null to x	file#	block#	le
global cache open s	file#	block#	le
global cache open x	file#	block#	le
global cache pending ast	le		
global cache pred cancel wait	le		
global cache recovery free wait			
global cache recovery quiesce wait	le		
global cache retry prepare	file#	block#	
global cache s to x	file#	block#	le
i/o slave wait	msg ptr		

Event Name	P1	P2	P3
imm op	msg ptr		
inactive session	session#	waited	
inactive transaction branch	branch#	waited	
index (re)build online cleanup	object	mode	wait
index (re)build online merge	object	mode	wait
index (re)build online start	object	mode	wait
index block split	rootdba	level	childdb
instance recovery	undo segment#		
instance state change	layer	value	waited
io done	msg ptr		
IPC busy async request			
IPC send completion sync	send count		
IPC wait for name service busy			
IPC waiting for OSD resources			
jobq slave wait			
KJC: Wait for msg sends to complete			
kjctcisnd: Queue/Send client message			
kjctssqmg: quick message send wait			
kksfbc child completion			
kksfbc research			
kkshgnc reloop			
kksscl hash split			
ksxr poll remote instances			
L1 validation	seghdr	l1bmb	
latch activity	address	number	process#
latch free	address	number	tries
LGWR random i/o			
LGWR sequential i/o			
LGWR wait for redo copy	copy latch #		
LGWR wait on ATTACH			
LGWR wait on DETACH			
LGWR wait on LNS			
LGWR wait on SENDREQ			
library cache load lock	object address	lock address	100*mask+namespace
library cache lock	handle address	lock address	100*mode+namespace
library cache pin	handle address	pin address	100*mode+namespace
listen endpoint status	end-point#	status	
LNS wait on ATTACH			
LNS wait on DETACH			
LNS wait on LGWR			
LNS wait on SENDREQ			
local write wait	file#	block#	
lock deadlock retry			

Wait Events and Parameters

Event Name	P1	P2	P3
lock escalate retry			
log buffer space			
log file parallel write	files	blocks	requests
log file sequential read	log#	block#	blocks
log file single write	log#	block#	blocks
log file switch (archiving needed)			
log file switch (checkpoint incomplete)			
log file switch (clearing log file)			
log file switch completion			
log file sync	buffer#		
log switch/archive	thread#		
multiple dbwriter suspend/resume for file offline			
name-service call wait	waittime		
null event			
parallel recovery coordinator waits for cleanup of slaves			
pending global transaction(s)	scans		
pi renounce write complete	file#	block#	
pipe get	handle address	buffer length	timeout
pipe put	handle address	record length	timeout
PL/SQL lock timer	duration		
pmon timer	duration		
process shutdown	type	process#	waited
process startup	type	process#	waited
PX create server	nserver	sleeptime	enqueue
PX Deq Credit: free buffer	sleeptime/senderid	passes	qref
PX Deq Credit: need buffer	sleeptime/senderid	passes	qref
PX Deq Credit: send blkd	sleeptime/senderid	passes	qref
PX Deq: Execute Reply	sleeptime/senderid	passes	
PX Deq: Execution Msg	sleeptime/senderid	passes	
PX Deq: Index Merge Close	sleeptime/senderid	passes	
PX Deq: Index Merge Execute	sleeptime/senderid	passes	
PX Deq: Index Merge Reply	sleeptime/senderid	passes	
PX Deq: Join ACK	sleeptime/senderid	passes	
PX Deq: kdcph_mai	kdcph_mai		
PX Deq: kdcphc_ack	kdcphc_ack		
PX Deq: Msg Fragment	sleeptime/senderid	passes	
PX Deq: OLAP Update Close	sleeptime/senderid	passes	
PX Deq: OLAP Update Execute	sleeptime/senderid	passes	
PX Deq: OLAP Update Reply	sleeptime/senderid	passes	
PX Deq: Par Recov Change Vector	sleeptime/senderid	passes	
PX Deq: Par Recov Execute	sleeptime/senderid	passes	
PX Deq: Par Recov Reply	sleeptime/senderid	passes	

Event Name	P1	P2	P3
PX Deq: Parse Reply	sleeptime/senderid	passes	
PX Deq: reap credit			
PX Deq: Signal ACK	sleeptime/senderid	passes	
PX Deq: Table Q Close	sleeptime/senderid	passes	
PX Deq: Table Q Get Keys	sleeptime/senderid	passes	
PX Deq: Table Q Normal	sleeptime/senderid	passes	
PX Deq: Table Q qref	sleeptime/senderid	passes	
PX Deq: Table Q Sample	sleeptime/senderid	passes	
PX Deq: Test for msg	sleeptime/senderid	passes	
PX Deq: Txn Recovery Reply	sleeptime/senderid	passes	
PX Deq: Txn Recovery Start	sleeptime/senderid	passes	
PX Deque wait	sleeptime/senderid	passes	
PX Idle Wait	sleeptime/senderid	passes	
PX Nsq: PQ descriptor query			
PX Nsq: PQ load info query			
PX qref latch	function	sleeptime	qref
PX Send Wait			
PX server shutdown	nalive	sleeptime	loop
PX signal server	serial	error	nbusy
PX slave connection			
PX slave release			
queue messages	queue id	process#	wait time
rdbms ipc message	timeout		
rdbms ipc message block			
rdbms ipc reply	from_process	timeout	
recovery read			
refresh controlfile command			
reliable message	channel context	channel handle	broadcast message
Replication Dequeue	sleeptime/senderid	passes	
resmgr:wait in actses run			
resmgr:waiting for cschlat1			
resmgr:waiting for cschlat2			
resmgr:waiting for cschlat3			
resmgr:waiting for cschlat4			
resmgr:waiting in check			
resmgr:waiting in check2			
resmgr:waiting in end wait			
resmgr:waiting in end wait2			
resmgr:waiting in enter			
resmgr:waiting in enter2			
resmgr:waiting in run (queued)			
resmgr:waiting in shutdown			
resmgr:waiting in system stop			

Wait Events and Parameters

Event Name	P1	P2	P3
retry contact SCN lock master			
RFS random i/o			
RFS sequential i/o			
row cache lock	cache id	mode	request
row cache read	cache id	address	times
sbtbackup			
sbtclose			
sbtclose2			
sbtcommand			
sbtend			
sbterror			
sbtinfo			
sbtinfo2			
sbtinit			
sbtinit2			
sbtopen			
sbtpcbbackup			
sbtppcancel			
sbtppcommit			
sbtppcend			
sbtppquerybackup			
sbtppqueryrestore			
sbtppcrestore			
sbtppcstart			
sbtppcstatus			
sbtppcvalidate			
sbtread			
sbtread2			
sbtremove			
sbtremove2			
sbtrestore			
sbtwrite			
sbtwrite2			
scginq AST call			
secondary event	event #	wait time	
select wait			
single-task message			
slave exit	nalive	sleeptime	loop
slave shutdown wait			
slave TJ process wait			
smon timer	sleep time	failed	
sort segment request			
SQL*Net break/reset to client	driver id	break?	not used

Event Name	P1	P2	P3
SQL*Net break/reset to dblink	driver id	break?	not used
SQL*Net message from client	driver id	#bytes	not used
SQL*Net message from dblink	driver id	#bytes	not used
SQL*Net message to client	driver id	#bytes	not used
SQL*Net message to dblink	driver id	#bytes	not used
SQL*Net more data from client	driver id	#bytes	not used
SQL*Net more data from dblink	driver id	#bytes	not used
SQL*Net more data to client	driver id	#bytes	not used
SQL*Net more data to dblink	driver id	#bytes	not used
statement suspended, wait error to be cleared			
STREAMS apply coord waiting for slave message			
STREAMS apply slave waiting for coord message			
STREAMS capture process filter callback wait for ruleset			
STREAMS capture process waiting for archive log			
STREAMS fetch slave waiting for txns			
Streams: Waiting for DDL to apply	sleep time		
switch logfile command			
switch undo - offline			
timer in sksawat			
trace continue	delay time		
trace unfreeze			
trace writer flush			
trace writer I/O			
transaction	undo seg# slot#	wrap#	count
txn to complete			
unbound tx			
undo segment extension	segment#		
undo segment recovery	segment#	tx flags	
undo segment tx slot	segment#		
virtual circuit status	circuit#	status	
wait for a paralle reco to abort			
wait for a undo record			
wait for activate message			
wait for another txn - rollback to savepoint			
wait for another txn - txn abort			
wait for another txn - undo rcv abort			
wait for assert messages to be sent			
Wait for Dictionary Build to lock all tables			
wait for EMON to spawn			
wait for FMON to come up			
wait for Logical Standby Apply shutdown			
wait for master scn	waittime		

Event Name	P1	P2	P3
wait for membership change			
wait for membership synchronization			
wait for message ack			
wait for MTTR advisory state object			
wait for name service busy			
wait for possible quiesce finish			
wait for record update			
wait for rr lock release			
wait for scn from all nodes	waittime		
wait for split-brain resolution			
wait for stopper event to be increased			
wait for sync ack	cluinc	pending_nd	
wait for tmc2 to complete			
wait for transaction			
wait for unread message on broadcast channel	channel context	channel handle	
wait for unread message on multiple broadcast channels	channel context	channel handle count	
wait for verification ack	cluinc	pending_insts	
wait for votes			
wait list latch activity	address	number	process#
wait list latch free	address	number	tries
waiting to get CAS latch			
waiting to get RM CAS latch			
wakeup event for builder			
wakeup event for preparer			
wakeup event for reader			
wakeup time manager			
write complete waits	file#	block#	
writes stopped by instance recovery or database suspension	by thread#	our thread#	

Parameter Descriptions

This section describes some of the more common event parameters in more detail.

block#

This is the block number of the block for which Oracle needs to wait. The block number is relative to the start of the file. To find the object to which this block belongs, enter the following SQL statements:

```
select name, kind
from ext_to_obj_view
where file# = file#
      and lowb <= block#
      and highb >= block#;
```

blocks

The number of blocks that is being either read from or written to the file. The block size is dependent on the file type:

- Database files have a block size of `DB_BLOCK_SIZE`
- Logfiles and controlfiles have a block size that is equivalent to the physical block size of the platform

break?

If the value for this parameter equals 0, a reset was sent to the client. A nonzero value indicates that a break was sent to the client.

class

The class of the block describes how the contents of the block are used. For example, class 1 represents data block, and class 4 represents segment header.

dba

The initials "dba" represents the data block address, which consists of a file number and a block number.

driver id

The address of the disconnect function of the driver that is currently being used.

file#

The following query returns the name of the database file:

```
select *
from v$datafile
where file# = file#;
```

id1

The first identifier (*id1*) of the enqueue or global lock takes its value from P2 or P2RAW. The meaning of the identifier depends on the name (P1).

id2

The second identifier (*id2*) of the enqueue or global lock takes its value from P3 or P3RAW. The meaning of the identifier depends on the name (P1).

le

The relative index number into `V$GC_ELEMENT`.

mode

The *mode* is usually stored in the low order bytes of P1 or P1RAW and indicates the mode of the enqueue or global lock request. This parameter has one of the following values:

Table C-1 Lock Mode Values

Mode Value	Description
1	Null mode
2	Sub-Share
3	Sub-Exclusive
4	Share
5	Share/Sub-Exclusive
6	Exclusive

Use the following SQL statement to retrieve the name of the lock and the mode of the lock request:

```
select chr(bitand(p1,-16777216)/16777215)||
       chr(bitand(p1, 16711680)/65535) "Lock",
       bitand(p1, 65536) "Mode"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

name and type

The name or "type" of the enqueue or global lock can be determined by looking at the two high order bytes of P1 or P1RAW. The name is always two characters. Use the following SQL statement to retrieve the lock name.

```
select chr(bitand(p1,-16777216)/16777215)||
       chr(bitand(p1,16711680)/65535) "Lock"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

namespace

The name of the object namespace as it is displayed in V\$DB_OBJECT_CACHE view.

requests

The number of I/Os that are "requested." This differs from the number of blocks in that one request could potentially contain multiple blocks.

session#

The number of the inactive session. Use the following SQL statement to find more information about the session:

```
select *
from v$session
where sid = session#;
```

waited

This is the total amount of time the session has waited for this session to terminate.

Wait Event Descriptions

This section describes some of the more common Oracle events in more detail.

alter system set dispatchers

A session has issued a statement `ALTER SYSTEM SET DISPATCHERS = string` and is waiting for the dispatchers to get started.

Wait Time: The session will wait 1 / 100 of a second and check to see if the new dispatchers have started else the session will wait again

Parameter	Description
<i>waited</i>	Number of times that the session has waited 1 / 100 of a second

batched allocate scn lock request

A session is waiting on another process to allocate a system change number (SCN). If the foreground timed out waiting on a process to get the SCN, the foreground will get the SCN.

Wait Time: The wait time is 1 second on the assumption that an SCN allocation should normally need much less than that

Parameters: None

BFILE check if exists

The session waits to check if an external large object (LOB) exists.

Wait Time: The total elapsed time for the `exists` call

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE check if open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the `isopen` call

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE closure

The session waits for an external large object (LOB) to close.

Wait Time: The total elapsed time for the `close` call

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE get length

The session waits on a call to check the size of an external large object (LOB).

Wait Time: The total elapsed time for the call to check the LOB size

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE get name object

The session waits on a call to find or generate the external name of a external large object.

Wait Time: The total elapse time for **make external file name** to complete

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE get path object

The session is waiting on a call to find or generate the external path name of an external large object (LOB).

Wait Time: The total elapsed time for **make external path** to complete

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE internal seek

The session waits for a positioning call within the external large object (LOB) to complete.

Wait Time: The total elapse time for the **seek** to complete

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the **isopen** call

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

BFILE read

The session waits for a read from a external large object (LOB) to complete.

Wait Time: The total elapse time for the **read** to complete

Parameter	Description
<i>session#</i>	See " session# " on page C-12
<i>waited</i>	See " waited " on page C-12

buffer busy waits

Wait until a buffer becomes available. This event happens because a buffer is either being read into the buffer cache by another session (and the session is waiting for that read to complete) or the buffer is the buffer cache, but in a incompatible mode (that is, some other session is changing the buffer).

Wait Time: Normal wait time is 1 second. If the session was waiting for a buffer during the last wait, then the next wait will be 3 seconds.

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>id</i>	The buffer busy wait event is called from different places in the session

buffer deadlock

Oracle does not really wait on this event; the foreground only yields the CPU. Thus, the chances of catching this event are very low. This is not an application induced deadlock, but an assumed deadlock by the cache layer. The cache layer cannot get a buffer in a certain mode within a certain amount of time.

Wait Time: 0 seconds. The foreground process only yields the CPU and will usually be placed at the end of the CPU run queue.

Parameter	Description
<i>class</i>	See " class " on page C-11
<i>mode</i>	See " mode " on page C-11
<i>flag</i>	The flag points to the internal flags used by the session to get this block
<i>dba</i>	See " dba " on page C-11

buffer for checkpoint

The buffer could not be checkpointed, because some process is modifying it. This means that after the wait, the DBWR will scan the whole buffer cache again. This could happen during a database close or after a user does a local checkpoint. During this situation the database cannot be closed.

Wait Time: 1 second

Parameter	Description
<i>dba</i>	See " dba " on page C-11
<i>state</i>	State refers to the status of the buffer contents

Parameter	Description
<i>mode</i>	See "mode" on page C-11
<i>buffer#</i>	This is the index of the block in the buffer cache (V\$BH)

buffer latch

The session waits on the buffer hash chain latch. Primarily used in the dump routines.

Wait Time: 1 second

Parameter	Description
<i>latch addr</i>	The virtual address in the SGA where this latch is located. Use the following statement to find the name of this latch: <pre>select * from v\$latch a, v\$latchname b where addr = <i>latch addr</i> and a.latch# = b.latch#;</pre>
<i>chain#</i>	The index into array of buffer hash chains. When the chain is 0xffffffff, the foreground waits on the LRU latch.

buffer read retry

This event occurs only if the instance is mounted in shared mode (Oracle Real Application Cluster). During the read of the buffer, the contents changed. This means that either:

- The version number, dba, or the incarnation and sequence number stored in the block no longer match
- The checksum on the block does not match the checksum in the block

The block will be re-read (this may fail up to 3 times), then corruption is assumed and the corrupt block is dumped in the trace file.

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
<i>file#</i>	See "file#" on page C-11
<i>block#</i>	See "block#" on page C-10

checkpoint completed

A session waits for a checkpoint to complete. This could happen, for example, during a close database or a local checkpoint.

Wait Time: 5 seconds

Parameters: None

checkpoint range buffer not saved

During a range checkpoint operation a buffer was found that was not saved or written. Either:

- The session will wait on this event if the write batch is empty and it is the first time that the session waited on this event in the range checkpoint operation

- The current range checkpoint operation will be aborted and a new one will be started to complete the operation

Wait Time: 10 milliseconds

Parameters: None

control file parallel write

This event occurs while the session is writing physical blocks to all control files. This happens when:

- The session starts a control file transaction (to make sure that the control files are up to date in case the session crashes before committing the control file transaction)
- The session commits a transaction to a control file
- Changing a generic entry in the control file, the new value is being written to all control files

Wait Time: The wait time is the time it takes to finish all writes to all control files

Parameter	Description
<i>files</i>	The number of control files to which the session is writing
<i>blocks</i>	The number of blocks that the session is writing to the control file
<i>requests</i>	The number of I/O requests which the session wants to write

control file sequential read

Reading from the control file. This happens in many cases. For example, while:

- Making a backup of the controlfiles
- Sharing information (between instances) from the controlfile
- Reading other blocks from the controlfiles
- Reading the header block

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
<i>file#</i>	The controlfile from which the session is reading
<i>block#</i>	Block number in the controlfile from where the session starts to read. The block size is the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
<i>blocks</i>	The number of blocks that the session is trying to read

control file single write

This wait is signaled while the control file's shared information is written to disk. This is an atomic operation protected by an enqueue (CF), so that only one session at a time can write to the entire database.

Wait Time: The wait time is the elapsed time of the write

Parameter	Description
<i>file#</i>	This identifies the control file to which the session is currently writing
<i>block#</i>	Block number in the control file where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
<i>blocks</i>	The number of blocks that the session is trying to read

conversion file read

This event occurs during the creation of a Version 7 controlfile as part of converting a database to Version 7 from Version 6.

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
<i>block#</i>	Block number in the control file where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
<i>blocks</i>	The number of blocks that the session is trying to read

db file parallel read

This happens during recovery. Database blocks that need to be changed as part of recovery are read in parallel from the database.

Wait Time: Wait until all of the I/Os are completed

Parameter	Description
<i>files</i>	This indicates the number of files to which the session is reading
<i>blocks</i>	This indicates the total number of blocks to be read
<i>requests</i>	This indicates the total number of I/O requests, which will be the same as blocks

db file parallel write

This event occurs in the DBWR. It indicates that the DBWR is performing a parallel write to files and blocks. The parameter *requests* indicates the real number of I/Os that are being performed. When the last I/O has gone to disk, the wait ends.

Wait Time: Wait until all of the I/Os are completed

Parameter	Description
<i>files</i>	This indicates the number of files to which the session is writing
<i>blocks</i>	This indicates the total number of blocks to be written
<i>requests</i>	This indicates the total number of I/O requests, which will be the same as blocks

db file scattered read

Similar to **db file sequential read**, except that the session is reading multiple data blocks.

Wait Time: The wait time is the actual time it takes to do all of the I/Os

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>blocks</i>	The number of blocks that the session is trying to read from the <i>file#</i> starting at <i>block#</i>

db file sequential read

The session waits while a sequential read from the database is performed. This event is also used for rebuilding the control file, dumping datafile headers, and getting the database file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>blocks</i>	This is the number of blocks that the session is trying to read (should be 1)

db file single write

This event is used to wait for the writing of the file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>blocks</i>	This is the number of blocks that the session is trying to write in <i>file#</i> starting at <i>block#</i>

DFS db file lock

This event occurs only for the DBWR in the Oracle Real Application Cluster. Each DBWR of every instance holds a global lock on each file in shared mode. The instance that is trying to offline the file will escalate the global lock from shared to exclusive. This signals the other instances to synchronize their SGAs with the controlfile before the file can be taken offline. The name of this lock is **DF** (see [Appendix D, "Oracle Enqueue Names"](#) for more information).

Wait Time: 1 second in loop. The DBWR is waiting in a loop (sleep, check) for the other instances to downgrade to NULL mode. During this time, the DBWR cannot perform other tasks such as writing buffers.

Parameter	Description
<i>file</i>	See " file# " on page C-11

DFS lock handle

The session waits for the lock handle of a global lock request. The lock handle identifies a global lock. With this lock handle, other operations can be performed on this global lock (to identify the global lock in future operations such as conversions or release). The global lock is maintained by the DLM.

Wait Time: The session waits in a loop until it has obtained the lock handle from the DLM. Inside the loop there is a wait of 0.5 seconds.

Parameter	Description
<i>name</i>	See " name and type " on page C-12
<i>mode</i>	See " mode " on page C-11
<i>id1</i>	See " id1 " on page C-11
<i>id2</i>	See " id2 " on page C-11

The session needs to get the lock handle.

direct path read

During Direct Path operations the data is asynchronously read from the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if during a direct read no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
<i>descriptor address</i>	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
<i>first dba</i>	The dba of the oldest I/O in the context referenced by the descriptor address
<i>block cnt</i>	Number of valid buffers in the context referenced by the descriptor address

direct path write

During Direct Path operations, the data is asynchronously written to the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if, during a direct write, no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
<i>descriptor address</i>	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting

Parameter	Description
<i>first dba</i>	The dba of the oldest I/O in the context referenced by the descriptor address
<i>block cnt</i>	Number of valid buffers in the context referenced by the descriptor address

dispatcher shutdown

During shutdown immediate or normal, the shutdown process must wait for all the dispatchers to shutdown. As each dispatcher is signaled, the session that causes the shutdown is waits on this event until the requested dispatcher is no longer alive.

Wait Time: 1 second

Parameter	Description
<i>waited</i>	Indicates the cumulative wait time. After 5 minutes, the session writes to the alert and trace files to indicate that there might be a problem.

dispatcher timer

This basically means that the dispatcher is idle and waiting for some work to arrive.

Wait Time: 60 seconds

Parameter	Description
<i>sleep time</i>	The intended sleep time. The dispatcher will return to work sooner if it is posted by either data arriving on the network or by a post from a shared server process to send data back to the client.

duplicate cluster key

It is possible for a race condition to occur when creating a new cluster key. If it is found that another process has put the cluster key into the data/index block, then the session waits and retries. The retry should then find a valid cluster key.

Wait Time: 0.01 seconds

Parameter	Description
<i>dba</i>	The dba of the block into which the session is trying to insert a cluster key

enqueue

The session is waiting for a local enqueue. The wait is dependent on the name of the enqueue (see [Appendix D, "Oracle Enqueue Names"](#)).

Wait Time: Depends on the enqueue name

Parameter	Description
<i>name</i>	See " name and type " on page C-12
<i>mode</i>	See " mode " on page C-11

file identify

The time it takes to identify a file so that it can be opened later.

file open

The time it takes to open the file.

free buffer waits

This will happen if:

- All buffer gets have been suspended. This could happen when a file was read-only and is now read-write. All the existing buffers need to be invalidated since they are not linked to lock elements (needed when mounted parallel (shared)). So cache buffers are not assigned to data block addresses until the invalidation is finished.
- The session moved some dirty buffers to the dirty queue and now this dirty queue is full. The dirty queue needs to be written first. The session will wait on this event and try again to find a free buffer
- This also happens after inspecting **free buffer inspected** buffers. If no free buffer is found, Oracle waits for one second, and then tries to get the buffer again (depends on the context). For more information, see [free buffer inspected](#) on page E-6.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10

free global transaction table entry

The session is waiting for a free slot in the global transaction table (used by the Distributed Database option). It will wait for 1 second and try again.

Wait Time: 1 second

Parameter	Description
<i>tries</i>	The number of times the session tried to find a free slot in the global transaction table

free process state object

Used during the creation of a process. The session will scan the process table and look for a free process slot. If none can be found, PMON is posted to check if all the processes currently in the process table are still alive. If there are dead processes, then PMON will clean them and make the process slot available to new processes. The waiting process will then rescan the process table to find the new slot.

Wait Time: 1 second

Parameters: None

global cache busy

The session waits to convert a buffer from Shared Current to Exclusive Current status.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>le</i>	See " lenum " on page A-8

global cache lock cleanup

PMON is waiting for an LCK process to cleanup the lock context after a foreground process died while doing a global cache lock operation.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>lenum</i>	See " lenum " on page A-8

global cache freelist wait

All releasable locks are used and a new one has been requested. To make a resource element available, a resource element is pinged.

Wait Time: The duration of the resource get operation to ping the resource element

Parameters: None

global cache null to s

The session waits for a resource convert from NULL to SHARED mode on the block identified by *file#* and *block#*.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>class</i>	See " class " on page C-11

global cache null to x

The session waits for a resource convert from NULL to EXCLUSIVE mode on the block identified by *file#* and *block#*.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>lenum</i>	See " le " on page C-11

global cache open s

The session waits for a resource get in `SHARED` mode on the block identified by `file#` and `block#`.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>class</i>	See " class " on page C-11

global cache open x

The session waits for a resource get in `EXCLUSIVE` mode on the block identified by `file#` and `block#`.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>lenum</i>	See " le " on page C-11

global cache s to x

The session waits for a resource convert from `SHARED` to `EXCLUSIVE` mode on the block identified by `file#` and `block#`.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	See " file# " on page C-11
<i>block#</i>	See " block# " on page C-10
<i>lenum</i>	See " le " on page C-11

inactive session

This event is used for two purposes:

- Switching sessions
If a time-out period has been specified, then wait that amount of time for the session to be detached.
- Killing sessions
From either `KILL SESSION` or internal request. Having posted a session that it should kill itself, wait for up to 1 minute for the session to terminate.

Wait Time: 1 second

Parameter	Description
<i>session#</i>	See " session# " on page C-12

Parameter	Description
<i>waited</i>	See "waited" on page C-12

inactive transaction branch

The session waits for a transaction branch that is currently used by another session.

Wait Time: 1 second

Parameter	Description
<i>branch#</i>	The serial number of the transaction for which the session is waiting
<i>waited</i>	See "waited" on page C-12

index block split

While trying to find an index key in an index block, Oracle noticed that the index block was being split. Oracle will wait for the split to finish and try to find the key again.

Wait Time: The session will yield the CPU, so there is no actual waiting time

Parameter	Description
<i>rootdba</i>	The root of the index
<i>level</i>	This is the level of the block that the session is trying to split in the index. The leaf blocks are level 0. If the level is > 0, it is a branch block. (The root block can be considered a special branch block).
<i>childdba</i>	The block that the session is trying to split

instance recovery

The session waits for SMON to finish the instance, transaction recovery, or sort segment cleanup.

Wait Time: The wait time can vary and depends on the amount of recovery needed

Parameter	Description
<i>undo segment#</i>	If the value is 0, SMON is probably performing instance recovery. If P1 > 0, use this query to find the undo segment: <pre>select * from v\$rollstat where usn = undo segment#;</pre>

instance state change

The session waits for SMON to enable or disable cache or transaction recovery. This usually happens during ALTER DATABASE OPEN or CLOSE.

Wait Time: Wait time depends on the amount of time the action takes (that is, the amount of recovery needed)

Parameter	Description
<i>layer</i>	This value can be 1 or 2. If 1, it means that the transaction layer wants transaction recovery to be performed. If 2, it means that cache recovery will be performed.

Parameter	Description
<i>value</i>	This value can be 0 (disable) or 1 (enable)
<i>waited</i>	The number of seconds waited so far

io done

The session waits for an I/O to complete or it waits for a slave process to become available to submit the I/O request. This event occurs on platforms that do not support asynchronous I/O.

Wait Time: 50 milliseconds

Parameter	Description
<i>msg ptr</i>	A pointer to the I/O request

kcl bg acks

The session waits for the background LCK process(es) to finish what they are doing. For example:

- Lock recovery
- Initializing the locks (start up)
- Finalizing the locks (shut down)

Wait Time: 10 seconds

Parameter	Description
<i>count</i>	The number of LCK processes that have finished
<i>loops</i>	The number times the process had to wait for the LCK processes to finish what they were doing

latch activity

This event is used as part of the process of determining whether a latch needs to be cleaned.

Wait Time: 0.05 to 0.1 seconds

Parameter	Description
<i>address</i>	The address of the latch that is being checked
<i>number</i>	The latch number of the latch that has activity. To find more information on the latch, use the following SQL statement: <pre>select * from v\$latchname where latch# = <i>number</i>;</pre>
<i>process#</i>	If this is 0, it is the first phase of the in-flux tests

latch free

The process waits for a latch that is currently busy (held by another process).

Wait Time: The wait time increases exponentially and does not include spinning on the latch (active waiting). The maximum wait time also depends on the number of latches that the process is holding. There is an incremental wait of up to 2 seconds.

Parameter	Description
<i>address</i>	The address of the latch for which the process is waiting
<i>number</i>	The latch number that indexes in the V\$LATCHNAME view. To find more information on the latch, use the following SQL statement: <pre>select * from v\$latchname where latch# = number;</pre>
<i>tries</i>	A count of the number of times the process tried to get the latch (slow with spinning) and the process has to sleep

library cache load lock

The session tries to find the load lock for the database object so that it can load the object. The load lock is always obtained in Exclusive mode, so that no other process can load the same object. If the load lock is busy the session will wait on this event until the lock becomes available.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
<i>object address</i>	Address of the object being loaded
<i>lock address</i>	Address of load lock being used
<i>mask</i>	Indicates which data pieces of the object that needs to be loaded

library cache lock

This event controls the concurrency between clients of the library cache. It acquires a lock on the object handle so that either:

- One client can prevent other clients from accessing the same object
- The client can maintain a dependency for a long time (for example, no other client can change the object)

This lock is also obtained to locate an object in the library cache.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
<i>handle address</i>	Address of the object being loaded
<i>lock address</i>	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is a State Object.
<i>mode</i>	Indicates the data pieces of the object which need to be loaded
<i>namespace</i>	See " namespace " on page C-12

library cache pin

This event manages library cache concurrency. Pinning an object causes the heaps to be loaded into memory. If a client wants to modify or examine the object, the client must acquire a pin after the lock.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
<i>handle address</i>	Address of the object being loaded
<i>pin address</i>	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is basically a State Object.
<i>mode</i>	Indicates which data pieces of the object that needs to be loaded
<i>namespace</i>	See " namespace " on page C-12

lock manager wait for remote message

The lock manager waits for a message from a remote lock manager in the same configuration.

Wait Time: The elapsed time of the wait

Parameter	Description
<i>waittime</i>	The elapsed time of the actual wait

log buffer space

Waiting for space in the log buffer because the session is writing data into the log buffer faster than LGWR can write it out. Consider making the log buffer bigger if it is small, or moving the log files to faster disks such as striped disks.

Wait Time: Usually 1 second, but 5 seconds if it is waiting for a Switch Logfile to complete

Parameters: None

log file parallel write

Writing redo records to the redo log files from the log buffer.

Wait Time: Time it takes for the I/Os to complete. Even though redo records are written in parallel, the parallel write is not complete until the last I/O is on disk.

Parameter	Description
<i>files</i>	Number of files to be written
<i>blocks</i>	Number of blocks to be written
<i>requests</i>	Number of I/O requests

log file sequential read

Waiting for the read from this logfile to return. This is used to read redo records from the log file.

Wait Time: Time it takes to complete the physical I/O (read)

Parameter	Description
<i>log#</i>	The relative sequence number of the logfiles within a log group (used only when dumping the logfiles)
<i>block#</i>	See " block# " on page C-10
<i>blocks</i>	The number of blocks to read

log file single write

Waiting for the write to this logfile to complete. This event is used while updating the header of the logfile. It is signaled when adding a log file member and when incrementing sequence numbers.

Wait Time: Time it takes for the physical I/O (write) to complete

Parameter	Description
<i>log#</i>	This is the number of the group/log to which the session is currently writing
<i>block#</i>	See " block# " on page C-10
<i>blocks</i>	The number of blocks to write

log file switch (archiving needed)

Waiting for a log switch because the log that the LGWR will be switching into has not been archived yet. Check the alert file to make sure that archiving has not stopped due to a failed archive write. To speed archiving, consider adding more archive processes or putting the archive files on striped disks.

Wait Time: 1 second

Parameters: None

log file switch (checkpoint incomplete)

Waiting for a log switch because the session cannot wrap into the next log. Wrapping cannot be performed because the checkpoint for that log has not completed.

Wait Time: 1 second

Parameters: None

log file switch (clearing log file)

Waiting for a log switch because the log is being cleared due to a `CLEAR LOGFILE` command or implicit clear logfile executed by recovery.

Wait Time: 1 second

Parameters: None

log file switch completion

Waiting for a log switch to complete.

Wait Time: 1 second

Parameters: None

log file sync

When a user session commits, the session's redo information needs to be flushed to the redo logfile. The user session will post the LGWR to write the log buffer to the redo log file. When the LGWR has finished writing, it will post the user session.

Wait Time: The wait time includes the writing of the log buffer and the post.

Parameter	Description
<i>buffer#</i>	The number of the physical buffer in the redo log buffer that needs to be synchronized

log switch/archive

Used as part of the ALTER SYSTEM ARCHIVE LOG CHANGE *scn* statement. The session waits for the current log from all open threads to be archived.

Wait Time: Wait for up to 10 seconds

Parameter	Description
<i>thread#</i>	The thread number of the thread that is currently archiving its current log

on-going SCN fetch to complete

Another session is fetching the SCN (system change number). This session waits for the other session finish fetching the SCN.

Wait Time: 1 second

Parameters: None

pending global transaction(s)

This event should happen only during testing. The session waits for pending transactions to clear.

Wait Time: 30 seconds

Parameter	Description
<i>scans</i>	Number of times the session has scanned the PENDING_TRANS\$ table

pipe get

The session waits for a message to be received on the pipe or for the pipe timer to expire.

Wait Time: There is a 5 second wake up (check) and the pipe timer set by the user

Parameter	Description
<i>handle address</i>	The library cache object handle for this pipe
<i>buffer length</i>	The length of the buffer
<i>timeout</i>	The pipe timer set by the user

pipe put

The session waits for the pipe send timer to expire or for space to be made available in the pipe.

Wait Time: There is the 5 second wakeup (check) and the user-supplied timeout value

Parameter	Description
<i>handle address</i>	The library cache object handle for this pipe
<i>record length</i>	The length of the record or buffer that has been put into the pipe
<i>timeout</i>	The pipe timer set by the user

PL/SQL lock timer

This event is called through the `DBMSLOCK.SLEEP` procedure or `USERLOCK.SLEEP` procedure. This event will most likely originate from procedures written by a user.

Wait Time: The wait time is in hundredths of seconds and is dependent on the user context

Parameter	Description
<i>duration</i>	The duration that the user specified in the <code>DBMS_LOCK.SLEEP</code> or <code>USER_LOCK.SLEEP</code> procedures

pmon timer

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

Wait Time: Up to 3 seconds, if not posted before

Parameter	Description
<i>duration</i>	The actual amount of time that the PMON is trying to sleep

process startup

Wait for a shared server, Dispatcher, or other background process to start.

Wait Time: Wait up to 1 second for a background process to start. If timed out, then re-wait until 5 minutes have passed and signal an error. If the process has started, the event will acknowledge this.

Parameter	Description
<i>type</i>	The process type that was started
<i>process#</i>	The process number of the process being started
<i>waited</i>	Cumulative time waited for the process to start

PX dequeue wait

The process is waiting for a message during a parallel execute.

Wait Time: The wait time depends on how quickly the message arrives. Wait times can vary, but it will normally be a short period of time.

Parameter	Description
<i>reason</i>	The reason for dequeuing
<i>sleeptime</i>	The amount of time that the session slept
<i>loop</i>	The total number of times that the session has slept

PX qref latch

Each parallel execution process has a parallel execution qref latch, which needs to be acquired before the queue buffers can be manipulated.

Wait Time: Wait up to 1 second

Parameter	Description
<i>function</i>	Indicates the type of wait that the session is doing
<i>sleeptime</i>	The amount of time that the session waits (in hundredths of a second)
<i>qref</i>	The address of the process queue for which the session is waits

PX server shutdown

During normal or immediate shutdown the parallel execution slaves are posted to shutdown cleanly. If any parallel execution slaves are still alive after 10 seconds, they are killed.

Wait Time: Wait up to 0.5 seconds

Parameter	Description
<i>nalive</i>	The number of parallel execution slaves that are still running
<i>sleeptime</i>	The total sleeptime since the session started to wait on this event
<i>loop</i>	The number of times the session waited for this event

PX signal server

This event occurs only in Exclusive mode. The query coordinator is signalling the Query Slaves that an error has occurred.

Wait Time: 0.5 seconds

Parameter	Description
<i>serial</i>	The serial number of the slave process queue
<i>error</i>	The error that has occurred
<i>nbusy</i>	The number of slave processes that are still busy

queue messages

The session is waiting on an empty OLTP queue (Advanced Queuing) for a message to arrive so that the session can dequeue that message.

Wait Time: The amount of time that the session wants to wait is determined by the parameter *wait time*

Parameter	Description
<i>queue id</i>	The ID of the OLTP queue for which this session is waiting
<i>process#</i>	The process number of the process in which this session runs
<i>wait time</i>	The intended wait time for this session

rdbms ipc message

The background processes (LGWR, DBWR, LMS0) use this event to indicate that they are idle and are waiting for the foreground processes to send them an IPC message to do some work.

Wait Time: Up to 3 seconds. The parameter *timeout* shows the true sleep time.

Parameter	Description
<i>timeout</i>	The amount of time that the session waits for an IPC message

rdbms ipc message block

This event indicates that all message blocks are in use and that the session had to wait for a message block to become available.

Wait Time: Wait up to 60 seconds

Parameters: None

rdbms ipc reply

This event is used to wait for a reply from one of the background processes.

Wait Time: The wait time is specified by the user and is indicated by the parameter *timeout*.

Parameter	Description
<i>from_process</i>	The background process for which the session is waiting. The wait is for a reply to an IPC message sent by the session.
<i>timeout</i>	The amount of time in seconds that this process will wait for a reply

redo wait

Defined but not used by the code.

row cache lock

The session is trying to get a data dictionary lock.

Wait Time: Wait up to 60 seconds.

Parameter	Description
<i>cache id</i>	The CACHE# column value in the V\$ROWCACHE view
<i>mode</i>	See " mode " on page C-11
<i>request</i>	The pipe timer set by the user

scginq AST call

Called by the session to find the highest lock mode that is held on a resource.

Wait Time: Wait up to 0.2 seconds, but the wait will continue until the NULL mode Acquisition AST has fired.

Parameters: None

single-task message

When running single task, this event indicates that the session waits for the client side of the executable.

Wait Time: Total elapsed time that this session spent in the user application

Parameters: None

smon timer

This is the main idle event for SMON. SMON will be waiting on this event most of the time until it times out or is posted by another process.

Wait Time: 5 minutes (300 seconds)

Parameter	Description
<i>sleeptime</i>	The amount of time that SMON tries to wait on this event in seconds
<i>failed</i>	The number of times SMON was posted when there some kind of error

SQL*Net break/reset to client

The server sends a break or reset message to the client. The session running on the server waits for a reply from the client.

Wait Time: The actual time it takes for the break or reset message to return from the client

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>break?</i>	See " break? " on page C-11

SQL*Net break/reset to dblink

Same as **SQL*Net break/reset to client**, but in this case, the break/reset message is sent to another server process over a database link.

Wait Time: The actual time it takes for the break or reset message to return from the other server process

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>break?</i>	See " break? " on page C-11

SQL*Net message from client

The server process (foreground process) waits for a message from the client process to arrive.

Wait Time: The time it took for a message to arrive from the client since the last message was sent to the client

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes received by the server (foreground process) from the client.

SQL*Net message from dblink

The session waits while the server process (foreground process) receives messages over a database link from another server process.

Wait Time: The time it took for a message to arrive from another server (foreground process) since a message was sent to the other foreground process.

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes received by the server (foreground process) from another foreground process over a database link.

SQL*Net message to client

The server (foreground process) is sending a message to the client.

Wait Time: The actual time the **send** takes

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes sent by the server process to the client

SQL*Net message to dblink

The server process (foreground process) is sending a message over a database link to another server process.

Wait Time: The actual time the **send** takes

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes sent by the server process to another server process over a database link

SQL*Net more data from client

The server is performing another send to the client. The previous operation was also a send to the client.

Wait Time: The time waited depends on the time it took to receive the data (including the waiting time)

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes received from the client

SQL*Net more data from dblink

The foreground process is expecting more data from a data base link.

Wait Time: The total time it takes to read the data from the database link (including the waiting time for the data to arrive)

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes received

SQL*Net more data to client

The server process is sending more data/messages to the client. The previous operation to the client was also a **send**.

Wait Time: The actual time it took for the **send** to complete

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes that are being sent to the client

SQL*Net more data to dblink

The event indicates that the server is sending data over a database link again. The previous operation over this database link was also a **send**.

Wait Time: The actual time it takes to send the data to the other server

Parameter	Description
<i>driver id</i>	See " driver id " on page C-11
<i>#bytes</i>	The number of bytes that are sent over the database link to the other server process

switch logfile command

The session waits on the user command SWITCH LOGFILE to complete.

Wait Time: 5 seconds

Parameters: None

timer in sksawat

The session waits for the Archiver (ARCH) asynchronous I/O to complete.

Wait Time: 0.01 seconds

Parameters: None

transaction

Wait for a blocking transaction to be rolled back. Continue waiting until the transaction has been rolled back.

Wait Time: 1 second

Parameter	Description
<i>undo seg#</i>	The rollback segment ID
<i>slot#</i>	The slot ID inside the rollback segment
<i>wrap#</i>	The sequence number that is incremented for each transaction
<i>count</i>	The number of times that the session has waited on this transaction

unbound tx

The session waits to see if there are any transactions that have been started but do not have a Rollback Segment associated with them.

Wait Time: 1 second

Parameters: None

undo segment extension

The undo segment is being extended or shrunk. The session must wait until the operation on the undo segment has finished.

Wait Time: 0.01 seconds

Parameter	Description
<i>segment#</i>	The ID of the rollback segment that is being extended or shrunk

undo segment recovery

PMON is rolling back a dead transaction. The wait continues until rollback finishes.

Wait Time: 3 seconds

Parameter	Description
<i>segment#</i>	The ID of the rollback segment that contains the transaction that is being rolled back
<i>tx flags</i>	The transaction flags (options) set for the transaction that is being rolled back

undo segment tx slot

Wait for a transaction slot to become available within the selected rollback segment. Continue waiting until the slot is available.

Wait Time: 1 second

Parameter	Description
<i>segment#</i>	The ID of the rollback segment that contains the transaction that is being rolled back

virtual circuit status

The session waits for a virtual circuit to return a message type indicated by *status*.

Wait Time: 30 seconds

Parameter	Description
<i>circuit#</i>	Indicates the virtual circuit# being waited on
<i>status</i>	Indicates what the session is waiting for

WMON goes to sleep

WMON is the UNIX-specific Wait Monitor, that can be used to reduce the number of system calls related to setting timers for posting or waiting in Oracle. You need to set an initialization parameter that enables the WMON process.

Wait Time: Depends on the next timeout

Parameters: None

write complete waits

The session waits for a buffer to be written. The write is caused by normal aging or by a cross-instance call.

Wait Time: 1 second

Parameter	Description
<i>file#</i>	The rollback segment id that contains the transaction that is being rolled back
<i>block#</i>	The transaction flags (options) set for the transaction that is being rolled back
<i>id</i>	Identifies the reason for waiting

writes stopped by instance recovery or database suspension

The session is blocked until the instance that started Instance Recovery is finished.

Wait Time: 5 seconds

Parameter	Description
<i>bythread#</i>	The rollback segment id that contains the transaction that is being rolled back
<i>ourthread#</i>	The current instance thread number

Oracle Enqueue Names

This appendix lists Oracle enqueues. **Enqueues** are shared memory structures (locks) that serialize access to database resources. They can be associated with a session or transaction. Enqueue names are displayed in the `LOCK_TYPE` column of the `DBA_LOCK` and `DBA_LOCK_INTERNAL` data dictionary views.

A **resource** uniquely identifies an object that can be locked by different sessions within an instance (local resource) or between instances (global resource). Each session that tries to lock the resource will have an **enqueue** on the resource.

Note: The names of enqueues and their definitions may change from release to release.

See Also: ["DBA_LOCK_INTERNAL"](#) on page 3-87 and ["DBA_LOCK"](#) on page 3-87

The Oracle enqueues are:

- BL, Buffer Cache Management
- BR, Backup/Restore
- CF, Controlfile Transaction
- CI, Cross-instance Call Invocation
- CU, Bind Enqueue
- DF, Datafile
- DL, Direct Loader Index Creation
- DM, Database Mount
- DR, Distributed Recovery Process
- DX, Distributed Transaction
- FS, File Set
- HW, High-Water Lock
- IN, Instance Number
- IR, Instance Recovery
- IS, Instance State
- IV, Library Cache Invalidation

-
- JI, Enqueue used during AJV snapshot refresh
 - JQ, Job Queue
 - KK, Redo Log "Kick"
 - L[A-P], Library Cache Lock
 - LS, Log Start or Switch
 - MM, Mount Definition
 - MR, Media Recovery
 - N[A-Z], Library Cache Pin
 - PE, ALTER SYSTEM SET PARAMETER = VALUE
 - PF, Password File
 - PI, Parallel Slaves
 - PR, Process Startup
 - PS, Parallel Slave Synchronization
 - Q[A-Z], Row Cache
 - RO, Object Reuse
 - RT, Redo Thread
 - RW, Row Wait
 - SC, System Commit Number
 - SM, SMON
 - SN, Sequence Number
 - SQ, Sequence Number Enqueue
 - SR, Synchronized Replication
 - SS, Sort Segment
 - ST, Space Management Transaction
 - SV, Sequence Number Value
 - TA, Transaction Recovery
 - TC, Thread Checkpoint
 - TE, Extend Table
 - TM, DML Enqueue
 - TO, Temporary Table Object Enqueue
 - TS, Temporary Segment (also TableSpace)
 - TT, Temporary Table
 - TX, Transaction
 - UL, User-defined Locks
 - UN, User Name
 - US, Undo Segment, Serialization
 - WL, Being Written Redo Log

-
- XA, Instance Attribute Lock
 - XI, Instance Registration Lock

Statistics Descriptions

This appendix describes the statistics stored in the V\$SESSTAT and V\$SYSSTAT dynamic performance tables. These statistics are useful in identifying and correcting performance problems.

This appendix contains the following topics:

- [Displaying Statistics](#)
- [Statistics Descriptions](#)

Displaying Statistics

The V\$SESSTAT view displays statistics on a per-session basis and is valid only for the session currently connected. When a session disconnects, all statistics for the session are updated in V\$SYSSTAT. The values for the statistics are cleared until the next session uses them.

The V\$STATNAME view contains all of the statistics for an Oracle release.

Many of these statistics are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice, even between patch releases.

Application developers should be aware of this and write their code to tolerate missing or extra statistics.

See Also: "[V\\$SESSTAT](#)" on page 5-47, "[V\\$STATNAME](#)" on page 5-71, and "[V\\$SYSSTAT](#)" on page 5-80 for more information on these views

Statistics Descriptions

This section describes the statistics stored in the V\$SESSTAT and V\$SYSSTAT views. The statistics are listed here in alphabetical order.

The CLASS column contains a number representing one or more statistics classes. The following class numbers are additive:

- 1, User
- 2, Redo
- 4, Enqueue
- 8, Cache

- 16, OS
- 32, Real Application Clusters
- 64, SQL
- 128, Debug

For example, a class value of 72 represents a statistic that relates to SQL statements and caching.

Some statistics are populated only if the `TIMED_STATISTICS` initialization parameter is set to `true`. Those statistics are flagged in the right-hand column.

Table E-1 Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
background checkpoints completed	8	Number of checkpoints completed by the background process. This statistic is incremented when the background process successfully advances the thread checkpoint.	
background checkpoints started	8	Number of checkpoints started by the background process. This statistic can be larger than "background checkpoints completed" if a new checkpoint overrides an incomplete checkpoint or if a checkpoint is currently under way. This statistic includes only checkpoints of the redo thread. It does not include: <ul style="list-style-type: none"> ■ Individual file checkpoints for operations such as offline or begin backup ■ Foreground (user-requested) checkpoints (for example, performed by <code>ALTER SYSTEM CHECKPOINT LOCAL</code> statements) 	
background timeouts	128	This is a count of the times where a background process has set an alarm for itself and the alarm has timed out rather than the background process being posted by another process to do some work.	
branch node splits	128	Number of times an index branch block was split because of the insertion of an additional value	
buffer is not pinned count	72	Number of times a buffer was free when visited. Useful only for internal debugging purposes.	
buffer is pinned count	72	Number of times a buffer was pinned when visited. Useful only for internal debugging purposes.	
bytes received via SQL*Net from client	1	Total number of bytes received from the client over Oracle Net Services	
bytes received via SQL*Net from dblink	1	Total number of bytes received from a database link over Oracle Net Services	
bytes sent via SQL*Net to client	1	Total number of bytes sent to the client from the foreground processes	
bytes sent via SQL*Net to dblink	1	Total number of bytes sent over a database link	
Cached Commit SCN referenced	128	Useful only for internal debugging purposes	
calls to get snapshot scn: kcmgss	32	Number of times a snapshot system change number (SCN) was allocated. The SCN is allocated at the start of a transaction.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
calls to kcmgas	128	Number of calls to routine kcmgas to get a new SCN	
calls to kcmgcs	128	Number of calls to routine kcmgcs to get a current SCN	
calls to kcmgrs	128	Number of calls to routine kcmgrs to get a recent SCN	
change write time	8	Elapsed redo write time for changes made to CURRENT blocks in 10s of milliseconds. This statistic is populated only if TIME_STATISTICS is set to true.	
cleanouts and rollbacks - consistent read gets	128	Number of consistent gets that require both block rollbacks and block cleanouts. See Also: "consistent gets"	
cleanouts only - consistent read gets	128	Number of consistent gets that require only block cleanouts, no rollbacks. See Also: "consistent gets"	
cluster key scan block gets	64	Number of blocks obtained in a cluster scan	
cluster key scans	64	Number of cluster scans that were started	
cold recycle reads	8	Number of buffers that were read through the least recently used end of the recycle cache with fast aging strategy	
commit cleanout failures: block lost	8	Number of times Oracle attempted a cleanout at commit but could not find the correct block due to forced write, replacement, or switch CURRENT	
commit cleanout failures: buffer being written	8	Number of times Oracle attempted a cleanout at commit, but the buffer was currently being written	
commit cleanout failures: callback failure	8	Number of times the cleanout callback function returns FALSE	
commit cleanout failures: cannot pin	8	Total number of times a commit cleanout was performed but failed because the block could not be pinned	
commit cleanout failures: hot backup in progress	8	Number of times Oracle attempted block cleanout at commit during hot backup. The image of the block needs to be logged before the buffer can be made dirty.	
commit cleanout failures: write disabled	8	Number of times a cleanout block at commit was performed but the writes to the database had been temporarily disabled	
commit cleanouts	8	Total number of times the cleanout block at commit function was performed	
commit cleanouts successfully completed	8	Number of times the cleanout block at commit function completed successfully	
Commit SCN cached	128	Number of times the system change number of a commit operation was cached	
consistent changes	8	Number of times a user process has applied rollback entries to perform a consistent read on the block Work loads that produce a great deal of consistent changes can consume a great deal of resources. The value of this statistic should be small in relation to the "consistent gets" statistic.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
consistent gets	8	Number of times a consistent read was requested for a block. See Also: " consistent changes " and " session logical reads " statistics	
CPU used by this session	1	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added. A similar problem can exist in the reporting by the operating system, especially on systems that suffer from many context switches.	3
CPU used when call started	128	The CPU time used when the call is started See Also: " CPU used by this session "	3
CR blocks created	8	Number of CURRENT blocks cloned to create CR (consistent read) blocks. The most common reason for cloning is that the buffer is held in a incompatible mode.	
current blocks converted for CR	8	Number CURRENT blocks converted to CR state	
cursor authentications	128	Number of privilege checks conducted during execution of an operation	
data blocks consistent reads - undo records applied	128	Number of undo records applied to data blocks that have been rolled back for consistent read purposes	
db block changes	8	Closely related to " consistent changes ", this statistic counts the total number of changes that were part of an update or delete operation that were made to all blocks in the SGA. Such changes generate redo log entries and hence become permanent changes to the database if the transaction is committed. This approximates total database work. It statistic indicates the rate at which buffers are being dirtied (on a per-transaction or per-second basis, for example).	
db block gets	8	Number of times a CURRENT block was requested See Also: " consistent gets "	
DBWR buffers scanned	8	Total number of dirty and clean buffers Oracle looks at when scanning LRU sets for dirty buffers to clean. Divide by " DBWR lru scans " to find the average number of buffers scanned.	
DBWR checkpoint buffers written	8	Number of buffers that were written for checkpoints	
DBWR checkpoints	8	Number of times the DBWR was asked to scan the cache and write all blocks marked for a checkpoint or the end of recovery. This statistic is always larger than " background checkpoints completed ".	
DBWR cross instance writes	40	Real Application Clusters only: Number of blocks written to satisfy a lock request from another instance	
DBWR free buffers found	8	Number of clean buffers that DBWR found when it was requested to make free buffers. Divide by " DBWR make free requests " to find the average number of reusable buffers at the end of each LRU.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
DBWR lru scans	8	Number of times that DBWR scans the LRU queue looking for buffers to write. This count includes scans to fill a batch being written for another purpose (such as a checkpoint). This statistic is always greater than or equal to "DBWR make free requests".	
DBWR make free requests	8	Number of requests to DBWR to make some free buffers for the LRU	
DBWR revisited being-written buffer	8	Number of times that DBWR tried to save a buffer for writing and found that it was already in the write batch. This statistic measures the amount of "useless" work that DBWR had to do in trying to fill the batch. Many sources contribute to a write batch. If the same buffer from different sources is considered for adding to the write batch, then all but the first attempt will be "useless" because the buffer is already marked as being written.	
DBWR summed scan depth	8	The current scan depth (number of buffers examined by DBWR) is added to this statistic every time DBWR scans the LRU for dirty buffers. Divide by "DBWR lru scans" to find the average scan depth.	
DBWR transaction table writes	8	Number of rollback segment headers written by DBWR. This statistic indicates how many "hot" buffers were written, causing a user process to wait while the write completed.	
DBWR undo block writes	8	Number of rollback segment blocks written by DBWR	
DDL statements parallelized	32	Number of DDL statements that were executed in parallel	
deferred (CURRENT) block cleanout applications	128	Number of times cleanout records are deferred, piggyback with changes, always current get	
DFO trees parallelized	32	Number of times a serial execution plan was converted to a parallel plan	
dirty buffers inspected	8	Number of dirty buffers found by the user process while the it is looking for a buffer to reuse	
DML statements parallelized	32	Number of DML statements that were executed in parallel	
enqueue conversions	4	Total number of conversions of the state of table or row lock	
enqueue deadlocks	4	Total number of deadlocks between table or row locks in different sessions	
enqueue releases	4	Total number of table or row locks released	
enqueue requests	4	Total number of table or row locks acquired	
enqueue timeouts	4	Total number of table and row locks (acquired and converted) that timed out before they could complete	
enqueue waits	4	Total number of waits that occurred during an enqueue convert or get because the enqueue get was deferred	
exchange deadlocks	8	Number of times that a process detected a potential deadlock when exchanging two buffers and raised an internal, restartable error. Index scans are the only operations that perform exchanges.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED STATISTICS
execute count	64	Total number of calls (user and recursive) that executed SQL statements	
free buffer inspected	8	Number of buffers skipped over from the end of an LRU queue in order to find a reusable buffer. The difference between this statistic and "dirty buffers inspected" is the number of buffers that could not be used because they had a user, a waiter, or were being read or written, or because they were busy or needed to be written after rapid aging out.	
free buffer requested	8	Number of times a reusable buffer or a free buffer was requested to create or load a block	
global cache blocks corrupt	40	Real Application Clusters only: Number of blocks that encountered a corruption or checksum failure during interconnect	
global cache convert time	40	Real Application Clusters only: Total time elapsed during lock converts	
global cache convert timeouts	40	Number of times lock converts in the global cache timed out	
global cache converts	40	Number of lock converts in the global cache	
global cache cr block receive time	40	Total amount of time foreground processes waited for a CR block to be sent through the interconnect. This statistic divided by "global cache cr blocks received" = time waited per block.	
global cache cr block serve time	40	Total amount of time the BSP process took to construct consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = construction time per CR block.	
global cache cr blocks received	40	Total number of blocks received	
global cache cr blocks served	40	Total number of blocks constructed by the BSP process	
global cache cr requests blocked	40	Number of times foreground attempt to request a cr block and failed	
global cache cr timeouts	40	Number of times a foreground process requested a consistent-read (CR) block when the request timed out	
global cache defers	40	Number of times a lock was requested and the holder of the lock deferred the release	
global cache freelist waits	40	System configured with fewer lock elements than buffers. Number of times foreground has to wait for a lock element.	
global cache get time	40	Total time spent waiting. This divided by global cache gets = time waited per request.	
global cache gets	40	Number of locks acquired	
global cache cr block send time	40	Total time spent by the BSP process in sending constructed consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = send time per CR block.	
global cache cr block log flushes	40	Number of log flushes of the consistent-read block	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
global cache cr block log flush time	40	Total time spent by the BSP process in log flushes after sending a constructed consistent-read (CR) block. This statistic divided by "global cache cr blocks served" = log flush time per CR block.	
global cache prepare failures	40	Number of times a failure occurred during preparation for interconnect transfer	
global lock async converts	32	Total number of asynchronous global lock converts	
global lock async gets	32	Total number of asynchronous global lock gets	
global lock convert time	32	Total elapsed time in 10s of milliseconds of all synchronous (non-asynchronous) global lock converts	
global lock get time	32	Total elapsed time in 10s of milliseconds of all synchronous (non-asynchronous) global lock gets	
global lock releases	32	Total number of synchronous global lock releases	
global lock sync converts	32	Total number of synchronous global lock converts	
global lock sync gets	32	Total number of synchronous global lock gets	
hot buffers moved to head of LRU	8	When a hot buffer reaches the tail of its replacement list, Oracle moves it back to the head of the list to keep it from being reused. This statistic counts such moves.	
immediate (CR) block cleanout applications	128	Number of times cleanout records are applied immediately during consistent-read requests	
immediate (CURRENT) block cleanout applications	128	Number of times cleanout records are applied immediately during current gets. Compare this statistic with "deferred (CURRENT) block cleanout applications"	
index fast full scans (direct read)	64	Number of fast full scans initiated using direct read	
index fast full scans (full)	64	Number of fast full scans initiated for full segments	
index fast full scans (rowid ranges)	64	Number of fast full scans initiated with rowid endpoints specified	
instance recovery database freeze count	32	Number of times the database is frozen during instance recovery	
kcmccs called get current scn	32	Number of times the kernel got the CURRENT SCN when there was a need to casually confirm the SCN	
kcmgss read scn without going to DLM	32	Number of times the kernel got a snapshot SCN without going to the distributed lock manager (DLM)	
kcmgss waited for batching	32	Number of times a database process is blocked waiting for a snapshot SCN	
leaf node splits	128	Number of times an index leaf node was split because of the insertion of an additional value	
logons cumulative	1	Total number of logons since the instance started. Useful only in V\$SYSSTAT. It gives an instance overview of all processes that logged on.	
logons current	1	Total number of current logons. Useful only in V\$SYSSTAT.	
messages received	128	Number of messages sent and received between background processes	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
messages sent	128	Number of messages sent and received between background processes	
native hash arithmetic execute	64	Number of hash operations performed using native arithmetic rather than Oracle NUMBERS	
native hash arithmetic fail	64	Number of has operations performed using native arithmetic that failed, requiring the hash operation to be performed with Oracle NUMBERS	
next scns gotten without going to DLM	32	Number of system change numbers obtained without going to the distributed lock manager or server	
no buffer to keep pinned count	72	Number of times a visit to a buffer attempted, but the buffer was not found where expected. Like "buffer is not pinned count" and "buffer is pinned count" , this statistic is useful only for internal debugging purposes.	
no work - consistent read gets	128	Number consistent gets that require neither block cleanouts nor rollbacks. See Also: "consistent gets"	
opened cursors cumulative	1	In V\$SYSSTAT: Total number of cursors opened since the instance started. In V\$SESSTAT: Total number of cursors opened since the start of the session.	
opened cursors current	1	Total number of current open cursors	
opens of replaced files	8	Total number of files that had to be reopened because they were no longer in the process file cache	
opens requiring cache replacement	8	Total number of file opens that caused a current file in the process file cache to be closed	
OS All other sleep time	16	Time spent sleeping for reasons other than misses in the data segment (see "OS Data page fault sleep time"), kernel page faults (see "OS Kernel page fault sleep time"), misses in the text segment (see "OS Text page fault sleep time"), or waiting for an OS locking object (see "OS User lock wait sleep time"). An example of such a reason is expiration of quanta.	
OS Chars read and written	16	Number of bytes read and written	
OS Data page fault sleep time	16	Time spent sleeping due to misses in the data segment	
OS Input blocks	16	Number of read I/Os	
OS Involuntary context switches	16	Number of context switches that were enforced by the operating system	
OS Kernel page fault sleep time	16	Time spent sleeping due to OS kernel page faults	
OS Major page faults	16	Number of page faults that resulted in I/O	
OS Messages received	16	Number of messages received	
OS Messages sent	16	Number of messages sent	
OS Minor page faults	16	Number of page faults that did not result in an actual I/O	
OS Other system trap CPU time	16	Total amount of time to process system traps (as distinct from system calls)	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
OS Output blocks	16	Number of write I/Os	
OS Process heap size	16	Size of area in memory allocated by the process. Typically this represents memory obtained by way of malloc().	
OS Process stack size	16	Size of the process stack segment	
OS Signals received	16	Number of signals received	
OS Swaps	16	Number of swap pages	
OS System call CPU time	16	Total amount of time spent executing in system mode	
OS System calls	16	Number of system calls	
OS Text page fault sleep time	16	Time spent sleeping due to misses in the text segment	
OS User level CPU time	16	Total amount of time spent executing in user mode	
OS User lock wait sleep time	16	Total amount of time sleeping while waiting for an OS locking object	
OS Voluntary context switches	16	Number of voluntary context switches (for example, when a process gives up the CPU by a SLEEP() system call)	
OS Wait-cpu (latency) time	16	Time spent sleeping while waiting for a CPU to become available	
Parallel operations downgraded 1 to 25 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 25 to 50 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 50 to 75 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 75 to 99 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded to serial	32	Number of times parallel execution was requested but execution was serial because of insufficient parallel execution servers	
Parallel operations not downgraded	32	Number of times parallel execution was executed at the requested degree of parallelism	
parse count (hard)	64	Total number of parse calls (real parses). A hard parse is a very expensive operation in terms of memory use, because it requires Oracle to allocate a workheap and other memory structures and then build a parse tree.	
parse count (total)	64	Total number of parse calls (hard and soft). A soft parse is a check on an object already in the shared pool, to verify that the permissions on the underlying object have not changed.	
parse time cpu	64	Total CPU time used for parsing (hard and soft) in 10s of 3 milliseconds	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
parse time elapsed	64	Total elapsed time for parsing, in 10s of milliseconds. Subtract "parse time cpu" from the this statistic to determine the total waiting time for parse resources.	3
physical reads	8	Total number of data blocks read from disk. This number equals the value of "physical reads direct" plus all reads into buffer cache.	
physical reads direct	8	Number of reads directly from disk, bypassing the buffer cache. For example, in high bandwidth, data-intensive operations such as parallel query, reads of disk blocks bypass the buffer cache to maximize transfer rates and to prevent the premature aging of shared data blocks resident in the buffer cache.	
physical writes	8	Total number of data blocks written to disk. This number equals the value of "physical writes direct" plus all writes from buffer cache.	
physical writes direct	8	Number of writes directly to disk, bypassing the buffer cache (as in a direct load operation)	
physical writes non checkpoint	8	Number of times a buffer is written for reasons other than advancement of the checkpoint. Used as a metric for determining the I/O overhead imposed by setting the FAST_START_IO_TARGET parameter to limit recovery I/Os. Essentially this statistic measures the number of writes that would have occurred had there been no checkpointing. Subtracting this value from "physical writes" gives the extra I/O for checkpointing.	
pinned buffers inspected	8	Number of times a user process, when scanning the tail of the replacement list looking for a buffer to reuse, encountered a cold buffer that was pinned or had a waiter that was about to pin it. This occurrence is uncommon, because a cold buffer should not be pinned very often.	
prefetched blocks	8	Number of contiguous and noncontiguous blocks that were prefetched	
prefetched blocks aged out before use	8	Number of contiguous and noncontiguous blocks that were prefetched but aged out before use	
process last non-idle time	128	The last time this process executed	3
PX local messages recv'd	32	Number of local messages received for parallel execution within the instance local to the current session	
PX local messages sent	32	Number of local messages sent for parallel execution within the instance local to the current session	
PX remote messages recv'd	32	Number of remote messages received for parallel execution within the instance local to the current session	
PX remote messages sent	32	Number of remote messages sent for parallel execution within the instance local to the current session	
queries parallelized	32	Number of SELECT statements executed in parallel	
recovery array read time	8	Elapsed time of I/O during recovery	
recovery array reads	8	Number of reads performed during recovery	
recovery blocks read	8	Number of blocks read during recovery	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
recursive calls	1	Number of recursive calls generated at both the user and system level. Oracle maintains tables used for internal processing. When Oracle needs to make a change to these tables, it internally generates an internal SQL statement, which in turn generates a recursive call.	
recursive cpu usage	1	Total CPU time used by non-user calls (recursive calls). Subtract this value from "CPU used by this session" to determine how much CPU time was used by the user calls.	
redo blocks written	2	Total number of redo blocks written. This statistic divided by "redo writes" equals number of blocks per write.	
redo buffer allocation retries	2	Total number of retries necessary to allocate space in the redo buffer. Retries are needed either because the redo writer has fallen behind or because an event such as a log switch is occurring.	
redo entries	2	Number of times a redo entry is copied into the redo log buffer	
redo log space requests	2	Number of times the active log file is full and Oracle must wait for disk space to be allocated for the redo log entries. Such space is created by performing a log switch. Log files that are small in relation to the size of the SGA or the commit rate of the work load can cause problems. When the log switch occurs, Oracle must ensure that all committed dirty buffers are written to disk before switching to a new log file. If you have a large SGA full of dirty buffers and small redo log files, a log switch must wait for DBWR to write dirty buffers to disk before continuing. Also examine the log file space and log file space switch wait events in V\$SESSION_WAIT	
redo log space wait time	2	Total elapsed waiting time for "redo log space requests" in 10s of milliseconds	3
redo log switch interrupts	2	Number of times that another instance asked this instance to advance to the next log file	
redo ordering marks	2	Number of times that a system change number was allocated to force a redo record to have a higher SCN than a record generated in another thread using the same block	
redo size	2	Total amount of redo generated in bytes	
redo synch time	8	Elapsed time of all "redo synch writes" calls in 10s of milliseconds	3
redo synch writes	8	Number of times a change being applied to the log buffer must be written out to disk due to a commit. The log buffer is a circular buffer that LGWR periodically flushes. Usually, redo that is generated and copied into the log buffer need not be flushed out to disk immediately.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
redo wastage	2	Number of bytes wasted because redo blocks needed to be written before they are completely full. Early writing may be needed to commit transactions, to be able to write a database buffer, or to switch logs.	
redo write time	2	Total elapsed time of the write from the redo log buffer to the current redo log file in 10s of milliseconds	3
redo writer latching time	2	Elapsed time in 10s of milliseconds needed by LGWR to obtain and release each copy latch	3
redo writes	2	Total number of writes by LGWR to the redo log files. "redo blocks written" divided by this statistic equals the number of blocks per write	
remote instance undo block writes	40	Number of times this instance wrote a rollback segment so that another instance could read it	
remote instance undo header writes	40	Number of times this instance wrote a undo header block so that another instance could read it	
rollback changes - undo records applied	128	Number of undo records applied to user-requested rollback changes (not consistent-read rollbacks)	
rollbacks only - consistent read gets	128	Number of consistent gets that require only block rollbacks, no block cleanouts. See Also: " consistent gets "	
rows fetched via callback	64	Rows fetched via callback. Useful primarily for internal debugging purposes.	
serializable aborts	1	Number of times a SQL statement in a serializable isolation level had to abort	
session connect time	1	The connect time for the session in 10s of milliseconds. This value is useful only in V\$SESSTAT. It is the wall clock time since the logon to this session occurred.	3
session cursor cache count	64	Total number of cursors cached. This statistic is incremented only if SESSION_CACHED_CURSORS > 0. This statistic is the most useful in V\$SESSTAT. If the value for this statistic in V\$SESSTAT is close to the setting of the SESSION_CACHED_CURSORS parameter, the value of the parameter should be increased.	
session cursor cache hits	64	Number of hits in the session cursor cache. A hit means that the SQL statement did not have to be reparsed. Subtract this statistic from " parse count (total) " to determine the real number of parses that occurred.	
session logical reads	1	The sum of " db block gets " plus " consistent gets "	
session pga memory	1	Current PGA size for the session. Useful only in V\$SESSTAT; it has no meaning in V\$SYSSTAT.	
session pga memory max	1	Peak PGA size for the session. Useful only in V\$SESSTAT; it has no meaning in V\$SYSSTAT.	
session stored procedure space	1	Amount of memory this session is using for stored procedures	
session uga memory	1	Current UGA size for the session. Useful only in V\$SESSTAT; it has no meaning in V\$SYSSTAT.	
session uga memory max	1	Peak UGA size for a session. Useful only in V\$SESSTAT; it has no meaning in V\$SYSSTAT.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
sorts (disk)	64	<p>Number of sort operations that required at least one disk write</p> <p>Sorts that require I/O to disk are quite resource intensive. Try increasing the size of the initialization parameter <code>SORT_AREA_SIZE</code>. For more information, see "SORT_AREA_SIZE" on page 1-131.</p>	
sorts (memory)	64	<p>Number of sort operations that were performed completely in memory and did not require any disk writes</p> <p>You cannot do much better than memory sorts, except maybe no sorts at all. Sorting is usually caused by selection criteria specifications within table join SQL operations.</p>	
sorts (rows)	64	Total number of rows sorted	
SQL*Net roundtrips to/from client	1	Total number of Net8 messages sent to and received from the client	
SQL*Net roundtrips to/from dblink	1	Total number of Net8 messages sent over and received from a database link	
summed dirty queue length	8	The sum of the dirty LRU queue length after every write request. Divide by write requests to get the average queue length after write completion.	
switch current to new buffer	8	Number of times the CURRENT block moved to a different buffer, leaving a CR block in the original buffer	
table fetch by rowid	64	<p>Number of rows that are fetched using a ROWID (usually recovered from an index)</p> <p>This occurrence of table scans usually indicates either non-optimal queries or tables without indexes. Therefore, this statistic should increase as you optimize queries and provide indexes in the application.</p>	
table fetch continued row	64	<p>Number of times a chained or migrated row is encountered during a fetch</p> <p>Retrieving rows that span more than one block increases the logical I/O by a factor that corresponds to the number of blocks that need to be accessed. Exporting and re-importing may eliminate this problem. Evaluate the settings for the storage parameters <code>PCTFREE</code> and <code>PCTUSED</code>. This problem cannot be fixed if rows are larger than database blocks (for example, if the <code>LONG</code> datatype is used and the rows are extremely large).</p>	
table scan blocks gotten	64	<p>During scanning operations, each row is retrieved sequentially by Oracle. This statistic counts the number of blocks encountered during the scan.</p> <p>This statistic tells you the number of database blocks that you had to get from the buffer cache for the purpose of scanning. Compare this value with the value of "consistent gets" to determine how much of the consistent read activity can be attributed to scanning.</p>	
table scan rows gotten	64	Number of rows that are processed during scanning operations	
table scans (cache partitions)	64	Number of range scans performed on tables that have the <code>CACHE</code> option enabled	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
table scans (direct read)	64	Number of table scans performed with direct read (bypassing the buffer cache)	
table scans (long tables)	64	Long (or conversely short) tables can be defined as tables that do not meet the short table criteria as described in table scans (short tables)	
table scans (rowid ranges)	64	During parallel query, the number of table scans conducted with specified ROWID ranges	
table scans (short tables)	64	Long (or conversely short) tables can be defined by optimizer hints coming down into the row source access layer of Oracle. The table must have the CACHE option set.	
total file opens	8	Total number of file opens performed by the instance. Each process needs a number of files (control file, log file, database file) in order to work against the database.	
transaction lock background get time	128	Useful only for internal debugging purposes	
transaction lock background gets	128	Useful only for internal debugging purposes	
transaction lock foreground requests	128	Useful only for internal debugging purposes	
transaction lock foreground wait time	128	Useful only for internal debugging purposes	
transaction rollbacks	128	Number of transactions being successfully rolled back	
transaction tables consistent read rollbacks	128	Number of times rollback segment headers are rolled back to create consistent read blocks	
transaction tables consistent reads - undo records applied	128	Number of undo records applied to transaction tables that have been rolled back for consistent read purposes	
Unnecessary process cleanup for SCN batching	32	Total number of times that the process cleanup was performed unnecessarily because the session or process did not get the next batched SCN. The next batched SCN went to another session instead.	
user calls	1	Number of user calls such as login, parse, fetch, or execute When determining activity, the ratio of user calls to RPI calls, give you an indication of how much internal work gets generated as a result of the type of requests the user is sending to Oracle.	
user commits	1	Number of user commits. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.	

Table E-1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
user rollbacks	1	Number of times users manually issue the ROLLBACK statement or an error occurs during a user's transactions	
write clones created in background	8	Number of times a background or foreground process clones a CURRENT buffer that is being written. The clone becomes the new, accessible CURRENT buffer, leaving the original buffer (now the clone) to complete writing.	
write clones created in foreground	8	Number of times a background or foreground process clones a CURRENT buffer that is being written. The clone becomes the new, accessible CURRENT buffer, leaving the original buffer (now the clone) to complete writing.	

A

- ACTIVE_INSTANCE_COUNT initialization parameter, 1-17
- ALL_ALL_TABLES view, 2-4
- ALL_APPLY view, 2-7
- ALL_APPLY_CONFLICT_COLUMNS view, 2-8
- ALL_APPLY_DML_HANDLERS view, 2-8
- ALL_APPLY_ENQUEUE view, 2-9
- ALL_APPLY_ERROR view, 2-9
- ALL_APPLY_EXECUTE view, 2-10
- ALL_APPLY_KEY_COLUMNS view, 2-10
- ALL_APPLY_PARAMETERS view, 2-10
- ALL_APPLY_PROGRESS view, 2-11
- ALL_APPLY_TABLE_COLUMNS view, 2-11
- ALL_ARGUMENTS view, 2-12
- ALL_ASSOCIATIONS view, 2-13
- ALL_AUDIT_POLICIES view, 2-14
- ALL_AUDIT_POLICY_COLUMNS view, 2-14
- ALL_AW_PS view, 2-15
- ALL_AWS view, 2-15
- ALL_BASE_TABLE_MVIEWS view, 2-16
- ALL_CAPTURE view, 2-16
- ALL_CAPTURE_EXTRA_ATTRIBUTES view, 2-18
- ALL_CAPTURE_PARAMETERS view, 2-18
- ALL_CAPTURE_PREPARED_DATABASE view, 2-19
- ALL_CAPTURE_PREPARED_SCHEMAS view, 2-19
- ALL_CAPTURE_PREPARED_TABLES view, 2-19
- ALL_CATALOG view, 2-20
- ALL_CLUSTER_HASH_EXPRESSIONS view, 2-20
- ALL_CLUSTERS view, 2-20
- ALL_COL_COMMENTS view, 2-21
- ALL_COL_PRIVS view, 2-22
- ALL_COL_PRIVS_MADE view, 2-22
- ALL_COL_PRIVS_RECD view, 2-23
- ALL_COLL_TYPES view, 2-23
- ALL_CONS_COLUMNS view, 2-24
- ALL_CONS_OBJ_COLUMNS view, 2-24
- ALL_CONSTRAINTS view, 2-25
- ALL_CONTEXT view, 2-26
- ALL_DB_LINKS view, 2-26
- ALL_DEF_AUDIT_OPTS view, 2-27
- ALL_DEPENDENCIES view, 2-28
- ALL_DIM_ATTRIBUTES view, 2-28
- ALL_DIM_CHILD_OF view, 2-29
- ALL_DIM_HIERARCHIES view, 2-29
- ALL_DIM_JOIN_KEY view, 2-29
- ALL_DIM_LEVEL_KEY view, 2-30
- ALL_DIM_LEVELS view, 2-30
- ALL_DIMENSIONS view, 2-31
- ALL_DIRECTORIES view, 2-31
- ALL_ERRORS view, 2-31
- ALL_EVALUATION_CONTEXT_TABLES view, 2-32
- ALL_EVALUATION_CONTEXT_VARS view, 2-33
- ALL_EVALUATION_CONTEXTS view, 2-33
- ALL_EXTERNAL_LOCATIONS view, 2-34
- ALL_EXTERNAL_TABLES view, 2-34
- ALL_IND_COLUMNS view, 2-35
- ALL_IND_EXPRESSIONS view, 2-36
- ALL_IND_PARTITIONS view, 2-36
- ALL_IND_STATISTICS view, 2-37
- ALL_IND_SUBPARTITIONS view, 2-39
- ALL_INDEXES view, 2-40
- ALL_INDEXTYPE_ARRAYTYPES view, 2-43
- ALL_INDEXTYPE_COMMENTS view, 2-43
- ALL_INDEXTYPE_OPERATORS view, 2-44
- ALL_INDEXTYPES view, 2-44
- ALL_INTERNAL_TRIGGERS view, 2-45
- ALL_JAVA_ARGUMENTS view, 2-45
- ALL_JAVA_CLASSES view, 2-46
- ALL_JAVA_DERIVATIONS view, 2-47
- ALL_JAVA_FIELDS view, 2-47
- ALL_JAVA_IMPLEMENTATIONS view, 2-48
- ALL_JAVA_INNERS view, 2-49
- ALL_JAVA_LAYOUTS view, 2-49
- ALL_JAVA_METHODS view, 2-50
- ALL_JAVA_NCOMPS view, 2-51
- ALL_JAVA_RESOLVERS view, 2-52
- ALL_JAVA_THROWS view, 2-52
- ALL_JOBS view, 2-53
- ALL_JOIN_IND_COLUMNS view, 2-54
- ALL_LIBRARIES view, 2-54
- ALL_LOB_PARTITIONS view, 2-54
- ALL_LOB_SUBPARTITIONS view, 2-55
- ALL_LOB_TEMPLATES view, 2-56
- ALL_LOBS view, 2-57
- ALL_LOG_GROUP_COLUMNS view, 2-58
- ALL_LOG_GROUPS view, 2-58
- ALL_METHOD_PARAMS view, 2-59

ALL_METHOD_RESULTS view, 2-59
 ALL_MVIEW_AGGREGATES view, 2-60
 ALL_MVIEW_ANALYSIS view, 2-61
 ALL_MVIEW_COMMENTS view, 2-62
 ALL_MVIEW_DETAIL_RELATIONS view, 2-62
 ALL_MVIEW_JOINS view, 2-63
 ALL_MVIEW_KEYS view, 2-63
 ALL_MVIEW_LOGS view, 2-64
 ALL_MVIEW_REFRESH_TIMES view, 2-65
 ALL_MVIEWS view, 2-65
 ALL_NESTED_TABLE_COLS view, 2-68
 ALL_NESTED_TABLES view, 2-70
 ALL_OBJ_COLATTRS view, 2-70
 ALL_OBJECT_TABLES view, 2-71
 ALL_OBJECTS view, 2-73
 ALL_OPANCILLARY view, 2-74
 ALL_OPARGUMENTS view, 2-74
 ALL_OPBINDINGS view, 2-75
 ALL_OPERATOR_COMMENTS view, 2-75
 ALL_OPERATORS view, 2-76
 ALL_OUTLINE_HINTS view, 2-76
 ALL_OUTLINES view, 2-77
 ALL_PART_COL_STATISTICS view, 2-77
 ALL_PART_HISTOGRAMS view, 2-78
 ALL_PART_INDEXES view, 2-79
 ALL_PART_KEY_COLUMNS view, 2-80
 ALL_PART_LOBS view, 2-81
 ALL_PART_TABLES view, 2-82
 ALL_PARTIAL_DROP_TABS view, 2-83
 ALL_PENDING_CONV_TABLES view, 2-83
 ALL_PLSQL_OBJECT_SETTINGS view, 2-84
 ALL_POLICIES view, 2-85
 ALL_POLICY_CONTEXTS view, 2-86
 ALL_POLICY_GROUPS view, 2-86
 ALL_PROCEDURES view, 2-87
 ALL_PROPAGATION view, 2-87
 ALL_PUBLISHED_COLUMNS view, 2-88
 ALL_QUEUE_TABLES view, 2-88
 ALL_QUEUES view, 2-89
 ALL_REFRESH view, 2-90
 ALL_REFRESH_CHILDREN view, 2-91
 ALL_REFRESH_DEPENDENCIES view, 2-91
 ALL_REFS view, 2-92
 ALL_REGISTERED_MVIEWS view, 2-92
 ALL_REGISTRY_BANNERS view, 2-93
 ALL_REPCAT view, 2-2
 ALL_REPCAT_REFRESH_TEMPLATES view, 2-2
 ALL_REPCAT_TEMPLATE_OBJECTS view, 2-2
 ALL_REPCAT_TEMPLATE_PARMS view, 2-2
 ALL_REPCAT_TEMPLATE_SITES view, 2-2
 ALL_REPCAT_USER_AUTHORIZATIONS view, 2-2
 ALL_REPCAT_USER_PARM_VALUES view, 2-2
 ALL_REPCATALOG view, 2-2
 ALL_REPCOLUMN view, 2-2
 ALL_REPCOLUMN_GROUP view, 2-2
 ALL_REPCONFLICT view, 2-2
 ALL_REPDDL view, 2-2
 ALL_REPFLAVOR_COLUMNS view, 2-2
 ALL_REPFLAVOR_OBJECTS view, 2-2
 ALL_REPFLAVORS view, 2-2
 ALL_REPGENERATED view, 2-2
 ALL_REPGENOBJECTS view, 2-2
 ALL_REPGROUP view, 2-2
 ALL_REPGROUP_PRIVILEGES view, 2-2
 ALL_REPGROUPED_COLUMN view, 2-2
 ALL_REPKKEY_COLUMNS view, 2-2
 ALL_REPOBJECT view, 2-2
 ALL_REPPARAMETER_COLUMN view, 2-2
 ALL_REPPRIORITY view, 2-2
 ALL_REPPRIORITY_GROUP view, 2-2
 ALL_REPPROP view, 2-2
 ALL_REPRESOL_STATS_CONTROL view, 2-2
 ALL_REPRESOLUTION view, 2-2
 ALL_REPRESOLUTION_METHOD view, 2-2
 ALL_REPRESOLUTION_STATISTICS view, 2-2
 ALL_REPSHEMA view, 2-3
 ALL_REPSITES view, 2-3
 ALL_REWRITE_EQUIVALENCES view, 2-93
 ALL_RULE_SET_RULES view, 2-93
 ALL_RULE_SETS view, 2-94
 ALL_RULES view, 2-94
 ALL_SCHEDULER_JOB_ARGS view, 2-95
 ALL_SCHEDULER_JOB_CLASSES view, 2-96
 ALL_SCHEDULER_JOB_LOG view, 2-96
 ALL_SCHEDULER_JOB_RUN_DETAILS view, 2-97
 ALL_SCHEDULER_JOBS view, 2-97
 ALL_SCHEDULER_PROGRAM_ARGS view, 2-99
 ALL_SCHEDULER_PROGRAMS view, 2-100
 ALL_SCHEDULER_RUNNING_JOBS view, 2-101
 ALL_SCHEDULER_SCHEDULES view, 2-101
 ALL_SCHEDULER_WINDOW_DETAILS view, 2-102
 ALL_SCHEDULER_WINDOW_GROUPS view, 2-102
 ALL_SCHEDULER_WINDOW_LOG view, 2-103
 ALL_SCHEDULER_WINDOWS view, 2-103
 ALL_SCHEDULER_WINDOWGROUP_MEMBERS view, 2-104
 ALL_SEC_RELEVANT_COLS view, 2-104
 ALL_SEQUENCES view, 2-105
 ALL_SERVICES view, 2-105
 ALL_SOURCE view, 2-106
 ALL_SOURCE_TABLES view, 2-106
 ALL_SQLJ_TYPE_ATTRS view, 2-107
 ALL_SQLJ_TYPE_METHODS view, 2-107
 ALL_SQLJ_TYPES view, 2-108
 ALL_STORED_SETTINGS view, 2-109
 ALL_STREAMS_GLOBAL_RULES view, 2-110
 ALL_STREAMS_MESSAGE_CONSUMERS view, 2-110
 ALL_STREAMS_MESSAGE_RULES view, 2-111
 ALL_STREAMS_NEWLY_SUPPORTED view, 2-111
 ALL_STREAMS_RULES view, 2-112
 ALL_STREAMS_SCHEMA_RULES view, 2-113
 ALL_STREAMS_TABLE_RULES view, 2-114
 ALL_STREAMS_TRANSFORM_FUNCTION view, 2-115
 ALL_STREAMS_UNSUPPORTED view, 2-116
 ALL_SUBPART_COL_STATISTICS view, 2-116

ALL_SUBPART_HISTOGRAMS view, 2-117
 ALL_SUBPART_KEY_COLUMNS view, 2-118
 ALL_SUBPARTITION_TEMPLATES view, 2-118
 ALL_SUBSCRIBED_COLUMNS view, 2-119
 ALL_SUBSCRIBED_TABLES view, 2-119
 ALL_SUBSCRIPTIONS view, 2-120
 ALL_SUMDELTA view, 2-120
 ALL_SYNONYMS view, 2-121
 ALL_TAB_COL_STATISTICS view, 2-121
 ALL_TAB_COLS view, 2-122
 ALL_TAB_COLUMNS view, 2-124
 ALL_TAB_COMMENTS view, 2-125
 ALL_TAB_HISTOGRAMS view, 2-126
 ALL_TAB_MODIFICATIONS view, 2-126
 ALL_TAB_PARTITIONS view, 2-127
 ALL_TAB_PRIVS view, 2-128
 ALL_TAB_PRIVS_MADE view, 2-129
 ALL_TAB_PRIVS_RECD view, 2-129
 ALL_TAB_STATISTICS view, 2-130
 ALL_TAB_SUBPARTITIONS view, 2-131
 ALL_TABLES view, 2-132
 ALL_TRIGGER_COLS view, 2-134
 ALL_TRIGGERS view, 2-135
 ALL_TYPE_ATTRS view, 2-135
 ALL_TYPE_METHODS view, 2-136
 ALL_TYPE_VERSIONS view, 2-137
 ALL_TYPES view, 2-137
 ALL_UNUSED_COL_TABS view, 2-138
 ALL_UPDATABLE_COLUMNS view, 2-138
 ALL_USERS view, 2-139
 ALL_USTATS view, 2-139
 ALL_VARRAYS view, 2-140
 ALL_VERSION_HVIEW view, 2-3
 ALL_VIEWS view, 2-140
 ALL_WARNING_SETTINGS view, 2-141
 ALL_WM_LOCKED_TABLES view, 2-3
 ALL_WM_MODIFIED_TABLES view, 2-3
 ALL_WM_RIC_INFO view, 2-3
 ALL_WM_TAB_TRIGGERS view, 2-3
 ALL_WM_VERSIONED_TABLES view, 2-3
 ALL_WM_VT_ERRORS view, 2-3
 ALL_WORKSPACE_PRIVS view, 2-3
 ALL_WORKSPACE_SAVEPOINTS view, 2-3
 ALL_WORKSPACES view, 2-3
 ALL_XML_SCHEMAS view, 2-142
 ALL_XML_TAB_COLS view, 2-142
 ALL_XML_TABLES view, 2-143
 ALL_XML_VIEW_COLS view, 2-143
 ALL_XML_VIEWS view, 2-144
 ALLOCATE EXTENT clause
 of ALTER TABLE
 instance number, 1-62
 allocation
 PCM locks, 1-56
 ALTER DATABASE statement
 ADD LOGFILE, 1-138
 ENABLE THREAD, 1-138
 THREAD, 1-138
 ALTER TABLE statement
 allocating extents, 1-62

ANALYZE statement, 2-40, 2-68, 2-122, 2-124, 2-132,
 3-82, 3-95, 3-128, 3-129, 3-130, 3-171, 3-182, 3-184
 AQ_TM_PROCESSES initialization parameter, 1-18
 ARCHIVE LOG START statement
 automatic archiving, 1-70
 ARCHIVE_LAG_TARGET initialization
 parameter, 1-18
 archived redo logs
 destination file, 1-70
 storage device, 1-70, 1-74
 ARCHIVELOG mode, 1-70
 archiving destination, 1-70, 1-74
 ASM_DISKGROUPS initialization parameter, 1-19
 ASM_DISKSTRING initialization parameter, 1-20
 ASM_POWER_LIMIT initialization parameter, 1-20
 AUDIT_ACTIONS table, 2-144
 AUDIT_FILE_DEST initialization parameter, 1-21
 AUDIT_SYS_OPERATIONS initialization
 parameter, 1-21
 AUDIT_TRAIL initialization parameter, 1-21
 authenticating remote clients, 1-118
 authenticating users, 1-100

B

BACKGROUND_CORE_DUMP initialization
 parameter, 1-22
 BACKGROUND_DUMP_DEST initialization
 parameter, 1-22
 BACKUP_TAPE_IO_SLAVES initialization
 parameter, 1-23
 BITMAP_MERGE_AREA_SIZE initialization
 parameter, 1-24
 BLANK_TRIMMING initialization parameter, 1-24
 blocks
 PCM lock, 1-56
 redo log, 1-79
 size, 1-33, 1-35, 1-41, A-2

C

cache
 dictionary, 5-27, 5-28, 5-29, 5-30
 statistics, 1-79
 CAT synonym for USER_CATALOG view, 2-145
 CATALOG view, 2-145
 CATALOG.SQL script, B-1
 creating V\$ views, 4-1
 CATAUDIT.SQL script
 audit trail records created by, 3-160
 audit trails for connections and
 disconnections, 3-160
 auditing characteristics described by, 3-171
 CATBLOCK.SQL script, B-2
 CATCLUST.SQL script, B-1
 CATEXP7.SQL script, B-2
 CATHS.SQL script, B-2
 CATIO.SQL script, B-2
 CATJAVA.SQL script, B-5
 CATNOADT.SQL script, B-4

- CATNOAUD.SQL script, B-4
- CATNOHS.SQL script, B-4
- CATNOPRT.SQL script, B-4
- CATNOQUE.SQL script, B-4
- CATNORMN.SQL script, B-4
- CATNOSVM.SQL script, B-4
- CATNSNMP.SQL script, B-4
- CATOCTK.SQL script, B-2
- CATPROC.SQL script, B-1
- CATQUEUE.SQL script, B-2
- CATREP.SQL script, B-2
- CATRMAN.SQL script, B-2
- CHAINED_ROWS table, 2-145
- CHANGE_SETS view, 2-145
- CHANGE_SOURCES view, 2-146
- CHANGE_TABLES view, 2-147
- characters
 - numeric group separators, 1-90
- checkpoints
 - checkpoint interval, 1-79
 - statistics, 1-79
- CIRCUITS initialization parameter, 1-25
- CLU synonym for USER_CLUSTERS view, 2-147
- CLUSTER_DATABASE initialization
 - parameter, 1-25
- CLUSTER_DATABASE_INSTANCES initialization
 - parameter, 1-25
- CLUSTER_INTERCONNECTS initialization
 - parameter, 1-26
- COL view, 2-147
- COLS synonym for USER_TAB_COLUMNS
 - view, 2-147
- COMMIT_POINT_STRENGTH initialization
 - parameter, 1-27
- COMPATIBLE initialization parameter, 1-27
- contention
 - block-level, 5-93
- control files
 - names, 1-29
 - specifying, 1-29
- CONTROL_FILE_RECORD_KEEP_TIME
 - initialization parameter, 1-28
- CONTROL_FILES initialization parameter, 1-28
- CORE_DUMP_DEST initialization parameter, 1-29
- cost-based optimization, 1-100
 - favoring IN-list iterators, 1-99
- CPU_COUNT initialization parameter, 1-29
- CREATE TABLE statement
 - FREELIST GROUPS clause, 1-62
- CREATE_BITMAP_AREA_SIZE initialization
 - parameter, 1-30
- CREATE_STORED_OUTLINES initialization
 - parameter, 1-30
- creating
 - a database
 - setting block size, 1-33, 1-35, 1-41
- currency, 1-87
 - international currency symbol, 1-88
- CURSOR_SHARING initialization parameter, 1-31
- CURSOR_SPACE_FOR_TIME initialization

- parameter, 1-32
- cursors
 - OPEN_CURSORS initialization parameter, 1-95
 - shared pool, 1-128

D

- D0902000.SQL script, B-5
- data blocks
 - reading multiple, 1-39
 - size of, 1-33, 1-35, 1-41
- data dictionary, 2-1
 - cache, 5-27, 5-28, 5-29, 5-30
 - datafiles, 1-56
 - scripts, B-2
 - SQL scripts, B-1
 - tables, 2-2
 - views, 1-56
 - overview, 2-2
 - user views, 2-1
- database writer process (DBWR)
 - checkpoint, 1-79
- DATABASE_PROPERTIES view, 2-147
- databases
 - block size, A-2
 - database limits, A-1
 - default language, 1-89
 - resource limits, 1-119, 1-120
- datafiles
 - mapping blocks to PCM locks, 1-56
 - maximum number, 1-40
 - parameter, 1-40
 - unspecified for PCM locks, 1-56
- dates
 - language used for, 1-88, 1-92
 - setting SYSDATE, 1-56
 - TO_CHAR function, 1-87
 - TO_DATE function, 1-87
- DB_BLOCK_BUFFERS initialization parameter, 1-33
- DB_BLOCK_CHECKING initialization
 - parameter, 1-34
- DB_BLOCK_CHECKSUM initialization
 - parameter, 1-34
- DB_BLOCK_SIZE initialization parameter, 1-35
- DB_CACHE_ADVICE initialization parameter, 1-35
- DB_CACHE_SIZE initialization parameter, 1-36
- DB_CREATE_FILE_DEST initialization
 - parameter, 1-37
- DB_CREATE_ONLINE_LOG_DEST_n initialization
 - parameter, 1-37
- DB_DOMAIN initialization parameter, 1-38
- DB_FILE_MULTIBLOCK_READ_COUNT
 - initialization parameter, 1-38
- DB_FILE_NAME_CONVERT initialization
 - parameter, 1-39
- DB_FILES initialization parameter, 1-40
- DB_FLASHBACK_RETENTION_TARGET
 - initialization parameter, 1-41
- DB_KEEP_CACHE_SIZE initialization
 - parameter, 1-41

DB_NAME initialization parameter, 1-41
 DB_nK_CACHE_SIZE initialization parameter, 1-32
 DB_RECOVERY_FILE_DEST initialization parameter, 1-42
 DB_RECOVERY_FILE_DEST_SIZE initialization parameter, 1-43
 DB_RECYCLE_CACHE_SIZE initialization parameter, 1-43
 DB_UNIQUE_NAME initialization parameter, 1-43
 DB_WRITER_PROCESSES initialization parameter, 1-44
 DBA_2PC_NEIGHBORS view, 3-1
 DBA_2PC_PENDING view, 3-1
 DBA_ADVISOR_ACTIONS view, 3-2
 DBA_ADVISOR_COMMANDS view, 3-3
 DBA_ADVISOR_DEFINITIONS view, 3-3
 DBA_ADVISOR_FINDINGS view, 3-4
 DBA_ADVISOR_JOURNAL view, 3-4
 DBA_ADVISOR_LOG view, 3-5
 DBA_ADVISOR_OBJECT_TYPES view, 3-6
 DBA_ADVISOR_OBJECTS view, 3-6
 DBA_ADVISOR_PARAMETERS view, 3-7
 DBA_ADVISOR_RATIONALE view, 3-8
 DBA_ADVISOR_RECOMMENDATIONS view, 3-9
 DBA_ADVISOR_SQLA_REC_SUM view, 3-10
 DBA_ADVISOR_SQLA_WK_MAP view, 3-10
 DBA_ADVISOR_SQLA_WK_STMTS view, 3-11
 DBA_ADVISOR_SQLW_JOURNAL view, 3-12
 DBA_ADVISOR_SQLW_PARAMETERS view, 3-12
 DBA_ADVISOR_SQLW_STMTS view, 3-13
 DBA_ADVISOR_SQLW_SUM view, 3-14
 DBA_ADVISOR_SQLW_TABLES view, 3-15
 DBA_ADVISOR_SQLW_TEMPLATES view, 3-15
 DBA_ADVISOR_TASKS view, 3-16
 DBA_ADVISOR_TEMPLATES view, 3-17
 DBA_ADVISOR_USAGE view, 3-17
 DBA_ALERT_HISTORY view, 3-18
 DBA_ALL_TABLES view, 3-18
 DBA_APPLICATION_ROLES view, 3-19
 DBA_APPLY view, 3-19
 DBA_APPLY_CONFLICT_COLUMNS view, 3-19
 DBA_APPLY_DML_HANDLERS view, 3-19
 DBA_APPLY_ENQUEUE view, 3-19
 DBA_APPLY_ERROR view, 3-19
 DBA_APPLY_EXECUTE view, 3-19
 DBA_APPLY_INSTANTIATED_GLOBAL view, 3-20
 DBA_APPLY_INSTANTIATED_OBJECTS view, 3-20
 DBA_APPLY_INSTANTIATED_SCHEMAS view, 3-20
 DBA_APPLY_KEY_COLUMNS view, 3-20
 DBA_APPLY_PARAMETERS view, 3-21
 DBA_APPLY_PROGRESS view, 3-21
 DBA_APPLY_TABLE_COLUMNS view, 3-21
 DBA_AQ_AGENT_PRIVS view, 3-21
 DBA_AQ_AGENTS view, 3-21
 DBA_ASSOCIATIONS view, 3-22
 DBA_ATTRIBUTE_TRANSFORMATIONS view, 3-22
 DBA_AUDIT_EXISTS view, 3-22
 DBA_AUDIT_OBJECT view, 3-23
 DBA_AUDIT_POLICIES view, 3-25
 DBA_AUDIT_POLICY_COLUMNS view, 3-25
 DBA_AUDIT_SESSION view, 3-25
 DBA_AUDIT_STATEMENT view, 3-26
 DBA_AUDIT_TRAIL view, 3-27
 DBA_AW_PS view, 3-29
 DBA_AWS view, 3-29
 DBA_BASE_TABLE_MVIEWS view, 3-30
 DBA_BLOCKERS view, 3-30
 DBA_CAPTURE view, 3-30
 DBA_CAPTURE_EXTRA_ATTRIBUTES view, 3-30
 DBA_CAPTURE_PARAMETERS view, 3-30
 DBA_CAPTURE_PREPARED_DATABASE view, 3-30
 DBA_CAPTURE_PREPARED_SCHEMAS view, 3-30
 DBA_CAPTURE_PREPARED_TABLES view, 3-31
 DBA_CATALOG view, 3-31
 DBA_CLU_COLUMNS view, 3-31
 DBA_CLUSTER_HASH_EXPRESSIONS view, 3-31
 DBA_CLUSTERS view, 3-31
 DBA_COL_COMMENTS view, 3-31
 DBA_COL_PRIVS view, 3-31
 DBA_COLL_TYPES view, 3-32
 DBA_COMMON_AUDIT_TRAIL view, 3-32
 DBA_CONS_OBJ_COLUMNS view, 3-34
 DBA_CONSTRAINTS view, 3-34
 DBA_CONTEXT view, 3-34
 DBA_DATA_FILES view, 3-34
 DBA_DATAPUMP_JOBS view, 3-34
 DBA_DATAPUMP_SESSIONS view, 3-35
 DBA_DB_LINKS view, 3-35
 DBA_DDL_LOCKS view, 3-35
 DBA_DEPENDENCIES view, 3-36
 DBA_DIM_ATTRIBUTES view, 3-36
 DBA_DIM_CHILD_OF view, 3-36
 DBA_DIM_HIERARCHIES view, 3-36
 DBA_DIM_JOIN_KEY view, 3-36
 DBA_DIM_LEVEL_KEY view, 3-37
 DBA_DIM_LEVELS view, 3-37
 DBA_DIMENSIONS view, 3-37
 DBA_DIRECTORIES view, 3-37
 DBA_DML_LOCKS view, 3-37
 DBA_DMT_FREE_SPACE view, 3-38
 DBA_DMT_USED_EXTENTS view, 3-38
 DBA_ENABLED_AGGREGATIONS view, 3-38
 DBA_ENABLED_TRACES view, 3-38
 DBA_ERRORS view, 3-39
 DBA_EVALUATION_CONTEXT_TABLES view, 3-39
 DBA_EVALUATION_CONTEXT_VARS view, 3-39
 DBA_EVALUATION_CONTEXTS view, 3-39
 DBA_EXP_FILES view, 3-39
 DBA_EXP_OBJECTS view, 3-40
 DBA_EXP_VERSION view, 3-40
 DBA_EXTENTS view, 3-40
 DBA_EXTERNAL_LOCATIONS view, 3-41
 DBA_EXTERNAL_TABLES view, 3-41

DBA_FEATURE_USAGE_STATISTICS view, 3-41
 DBA_FGA_AUDIT_TRAIL view, 3-43
 DBA_FREE_SPACE view, 3-44
 DBA_FREE_SPACE_COALESCED view, 3-45
 DBA_HIGH_WATER_MARK_STATISTICS
 view, 3-45
 DBA_HIST_ACTIVE_SESS_HISTORY view, 3-46
 DBA_HIST_BASELINE view, 3-47
 DBA_HIST_BG_EVENT_SUMMARY view, 3-47
 DBA_HIST_BUFFER_POOL_STAT view, 3-47
 DBA_HIST_CR_BLOCK_SERVER view, 3-48
 DBA_HIST_CURRENT_BLOCK_SERVER
 view, 3-49
 DBA_HIST_DATABASE_INSTANCE view, 3-50
 DBA_HIST_DATAFILE view, 3-50
 DBA_HIST_DB_CACHE_ADVICE view, 3-50
 DBA_HIST_DLM_MISC view, 3-51
 DBA_HIST_ENQUEUE_STAT view, 3-51
 DBA_HIST_EVENT_NAME view, 3-52
 DBA_HIST_FILEMETRIC_HISTORY view, 3-52
 DBA_HIST_FILESTATXS view, 3-52
 DBA_HIST_INSTANCE_RECOVERY view, 3-53
 DBA_HIST_JAVA_POOL_ADVICE view, 3-54
 DBA_HIST_LATCH view, 3-55
 DBA_HIST_LATCH_CHILDREN view, 3-56
 DBA_HIST_LATCH_MISSES_SUMMARY
 view, 3-56
 DBA_HIST_LATCH_NAME view, 3-57
 DBA_HIST_LATCH_PARENT view, 3-57
 DBA_HIST_LIBRARYCACHE view, 3-57
 DBA_HIST_LOG view, 3-58
 DBA_HIST_METRIC_NAME view, 3-59
 DBA_HIST_MTRR_TARGET_ADVICE view, 3-59
 DBA_HIST_OPTIMIZER_ENV view, 3-60
 DBA_HIST_OSSTAT view, 3-60
 DBA_HIST_OSSTAT_NAME view, 3-61
 DBA_HIST_PARAMETER view, 3-61
 DBA_HIST_PARAMETER_NAME view, 3-61
 DBA_HIST_PGA_TARGET_ADVICE view, 3-61
 DBA_HIST_PGASTAT view, 3-62
 DBA_HIST_RESOURCE_LIMIT view, 3-63
 DBA_HIST_ROLLSTAT view, 3-63
 DBA_HIST_ROWCACHE_SUMMARY view, 3-64
 DBA_HIST_SEG_STAT view, 3-64
 DBA_HIST_SEG_STAT_OBJ view, 3-66
 DBA_HIST_SERVICE_NAME view, 3-66
 DBA_HIST_SERVICE_STAT view, 3-66
 DBA_HIST_SERVICE_WAIT_CLASS view, 3-67
 DBA_HIST_SESSMETRIC_HISTORY view, 3-67
 DBA_HIST_SGA view, 3-67
 DBA_HIST_SGASTAT view, 3-68
 DBA_HIST_SHARED_POOL_ADVICE view, 3-68
 DBA_HIST_SNAP_ERROR view, 3-69
 DBA_HIST_SNAPSHOT view, 3-69
 DBA_HIST_SQL_PLAN view, 3-69
 DBA_HIST_SQL_SUMMARY view, 3-71
 DBA_HIST_SQL_WORKAREA_HSTGRM
 view, 3-71
 DBA_HIST_SQLBIND view, 3-72
 DBA_HIST_SQLSTAT view, 3-72
 DBA_HIST_SQLTEXT view, 3-75
 DBA_HIST_STAT_NAME view, 3-75
 DBA_HIST_SYS_TIME_MODEL view, 3-75
 DBA_HIST_SYSMETRIC_HISTORY view, 3-75
 DBA_HIST_SYSMETRIC_SUMMARY view, 3-76
 DBA_HIST_SYSSTAT view, 3-76
 DBA_HIST_SYSTEM_EVENT view, 3-77
 DBA_HIST_TABLESPACE_STAT view, 3-77
 DBA_HIST_TBSPC_SPACE_USAGE view, 3-77
 DBA_HIST_TEMPFILE view, 3-78
 DBA_HIST_TEMPSTATXS view, 3-78
 DBA_HIST_THREAD view, 3-79
 DBA_HIST_UNDOSTAT view, 3-79
 DBA_HIST_WAITCLASSMET_HISTORY view, 3-80
 DBA_HIST_WAITSTAT view, 3-81
 DBA_HIST_WR_CONTROL view, 3-81
 DBA_IND_COLUMNS view, 3-81
 DBA_IND_EXPRESSIONS view, 3-81
 DBA_IND_PARTITIONS view, 3-82
 DBA_IND_STATISTICS view, 3-82
 DBA_IND_SUBPARTITIONS view, 3-82
 DBA_INDEXES view, 3-82
 collecting statistics for, 3-82
 DBA_INDEXTYPE_ARRAYTYPES view, 3-82
 DBA_INDEXTYPE_COMMENTS view, 3-82
 DBA_INDEXTYPE_OPERATORS view, 3-82
 DBA_INDEXTYPES view, 3-82
 DBA_INTERNAL_TRIGGERS view, 3-83
 DBA_JAVA_ARGUMENTS view, 3-83
 DBA_JAVA_CLASSES view, 3-83
 DBA_JAVA_DERIVATIONS view, 3-83
 DBA_JAVA_FIELDS view, 3-83
 DBA_JAVA_IMPLEMENTED view, 3-83
 DBA_JAVA_INNERS view, 3-83
 DBA_JAVA_LAYOUTS view, 3-83
 DBA_JAVA_METHODS view, 3-84
 DBA_JAVA_NCOMPS view, 3-84
 DBA_JAVA_POLICY view, 3-84
 DBA_JAVA_RESOLVERS view, 3-84
 DBA_JAVA_THROWS view, 3-84
 DBA_JOBS view, 3-85
 DBA_JOBS_RUNNING view, 3-85
 DBA_JOIN_IND_COLUMNS view, 3-85
 DBA_KGLLOCK view, 3-85
 DBA_LIBRARIES view, 3-85
 DBA_LMT_FREE_SPACE view, 3-86
 DBA_LMT_USED_EXTENTS view, 3-86
 DBA_LOB_PARTITIONS view, 3-86
 DBA_LOB_SUBPARTITIONS view, 3-86
 DBA_LOB_TEMPLATES view, 3-86
 DBA_LOBS view, 3-86
 DBA_LOCK view, 3-87
 DBA_LOCKS synonym, 3-87
 DBA_LOCK_INTERNAL view, 3-87
 DBA_LOCKS synonym for DBA_LOCK view, 3-87
 DBA_LOG_GROUP_COLUMNS view, 3-87
 DBA_LOG_GROUPS view, 3-87
 DBA_LOGMNR_LOG view, 3-88
 DBA_LOGMNR_PURGED_LOG view, 3-88
 DBA_LOGMNR_SESSION view, 3-88

DBA_LOGSTDBY_EVENTS view, 3-89
 DBA_LOGSTDBY_LOG view, 3-90
 DBA_LOGSTDBY_NOT_UNIQUE view, 3-90
 DBA_LOGSTDBY_PARAMETERS view, 3-91
 DBA_LOGSTDBY_PROGRESS view, 3-91
 DBA_LOGSTDBY_SKIP view, 3-92
 DBA_LOGSTDBY_SKIP_TRANSACTION view, 3-92
 DBA_LOGSTDBY_UNSUPPORTED view, 3-93
 DBA_METHOD_PARAMS view, 3-93
 DBA_METHOD_RESULTS view, 3-93
 DBA_MVIEW_AGGREGATES view, 3-93
 DBA_MVIEW_ANALYSIS view, 3-93
 DBA_MVIEW_COMMENTS view, 3-94
 DBA_MVIEW_DETAIL_RELATIONS view, 3-94
 DBA_MVIEW_JOINS view, 3-94
 DBA_MVIEW_KEYS view, 3-94
 DBA_MVIEW_LOG_FILTER_COLS view, 3-94
 DBA_MVIEW_LOGS view, 3-94
 DBA_MVIEW_REFRESH_TIMES view, 3-95
 DBA_MVIEWS view, 3-95
 DBA_NESTED_TABLE_COLS view, 3-95
 DBA_NESTED_TABLES view, 3-95
 DBA_OBJ_AUDIT_OPTS view, 3-95
 DBA_OBJ_COLATTRS view, 3-96
 DBA_OBJECT_SIZE view, 3-96
 DBA_OBJECT_TABLES view, 3-97
 DBA_OBJECTS view, 3-97
 DBA_OPANCILLARY view, 3-97
 DBA_OPARGUMENTS view, 3-97
 DBA_OPBINDINGS view, 3-97
 DBA_OPERATOR_COMMENTS view, 3-97
 DBA_OPERATORS view, 3-97
 DBA_ORPHAN_KEY_TABLE view, 3-97
 DBA_OUTLINE_HINTS view, 3-98
 DBA_OUTLINES view, 3-98
 DBA_OUTSTANDING_ALERTS view, 3-98
 DBA_PART_COL_STATISTICS view, 3-99
 DBA_PART_HISTOGRAMS view, 3-99
 DBA_PART_INDEXES view, 3-99
 DBA_PART_KEY_COLUMNS view, 3-99
 DBA_PART_LOBS view, 3-99
 DBA_PART_TABLES view, 3-99
 DBA_PARTIAL_DROP_TABS view, 3-100
 DBA_PENDING_CONV_TABLES view, 3-100
 DBA_PENDING_TRANSACTIONS view, 3-100
 DBA_PLSQL_OBJECT_SETTINGS view, 3-100
 DBA_POLICIES view, 3-100
 DBA_POLICY_CONTEXTS view, 3-100
 DBA_POLICY_GROUPS view, 3-100
 DBA_PRIV_AUDIT_OPTS view, 3-101
 DBA_PROCEDURES view, 3-101
 DBA_PROFILES view, 3-101
 DBA_PROPAGATION view, 3-101
 DBA_PROXIES view, 3-101
 DBA_PUBLISHED_COLUMNS view, 3-102
 DBA_QUEUE_SCHEDULES view, 3-102
 DBA_QUEUE_TABLES view, 3-103
 DBA_QUEUES view, 3-103
 DBA_RCHILD view, 3-103
 DBA_RECYCLEBIN view, 3-104
 DBA_REDEFINITION_ERRORS view, 3-105
 DBA_REDEFINITION_OBJECTS view, 3-105
 DBA_REFRESH view, 3-106
 DBA_REFRESH_CHILDREN view, 3-106
 DBA_REFS view, 3-106
 DBA_REGISTERED_ARCHIVED_LOG view, 3-106
 DBA_REGISTERED_MVIEW_GROUPS view, 3-106
 DBA_REGISTERED_MVIEWS view, 3-107
 DBA_REGISTRY view, 3-107
 DBA_REGISTRY_HIERARCHY view, 3-108
 DBA_REPAIR_TABLE view, 3-108
 DBA_REPCAT view, 2-2
 DBA_REPCAT_EXCEPTIONS view, 2-2
 DBA_REPCAT_REFRESH_TEMPLATES view, 2-2
 DBA_REPCAT_TEMPLATE_OBJECTS view, 2-2
 DBA_REPCAT_TEMPLATE_PARAMS view, 2-2
 DBA_REPCAT_TEMPLATE_SITES view, 2-2
 DBA_REPCAT_USER_AUTHORIZATIONS view, 2-2
 DBA_REPCAT_USER_PARM_VALUES view, 2-2
 DBA_REPCATLOG view, 2-2
 DBA_REPCOLUMN view, 2-2
 DBA_REPCOLUMN_GROUP view, 2-2
 DBA_REPCONFLICT view, 2-2
 DBA_REPDDL view, 2-2
 DBA_REPEXTENSIONS view, 2-2
 DBA_REPFLAVOR_COLUMNS view, 2-2
 DBA_REPFLAVOR_OBJECTS view, 2-2
 DBA_REPFLAVORS view, 2-2
 DBA_REPGENERATED view, 2-2
 DBA_REPGENOBJECTS view, 2-2
 DBA_REPGROUP view, 2-2
 DBA_REPGROUP_PRIVILEGES view, 2-2
 DBA_REPGROUPED_COLUMN view, 2-2
 DBA_REPKEY_COLUMNS view, 2-2
 DBA_REPOBJECT view, 2-2
 DBA_REPPARAMETER_COLUMN view, 2-2
 DBA_REPPRIORITY view, 2-2
 DBA_REPPRIORITY_GROUP view, 2-2
 DBA_REPPROP view, 2-2
 DBA_REPRESOL_STATS_CONTROL view, 2-2
 DBA_REPRESOLUTION view, 2-2
 DBA_REPRESOLUTION_METHOD view, 2-2
 DBA_REPRESOLUTION_STATISTICS view, 2-2
 DBA_REPSHEMA view, 2-3
 DBA_REPSITES view, 2-3
 DBA_REPSITES_NEW view, 2-3
 DBA_RESUMABLE view, 3-109
 DBA_REWRITE_EQUIVALENCES view, 3-110
 DBA_RGROUP view, 3-110
 DBA_ROLE_PRIVS view, 3-110
 DBA_ROLES view, 3-110
 DBA_ROLLBACK_SEGS view, 3-111
 DBA_RSRC_CONSUMER_GROUP_PRIVS view, 3-111
 DBA_RSRC_CONSUMER_GROUPS view, 3-112
 DBA_RSRC_GROUP_MAPPINGS view, 3-112
 DBA_RSRC_MANAGER_SYSTEM_PRIVS view, 3-112

DBA_RSRC_MAPPING_PRIORITY view, 3-113
 DBA_RSRC_PLAN_DIRECTIVES view, 3-113
 DBA_RSRC_PLANS view, 3-114
 DBA_RULE_SET_RULES view, 3-115
 DBA_RULE_SETS view, 3-115
 DBA_RULES view, 3-115
 DBA_SCHEDULER_JOB_ARGS view, 3-115
 DBA_SCHEDULER_JOB_CLASSES view, 3-115
 DBA_SCHEDULER_JOB_LOG view, 3-115
 DBA_SCHEDULER_JOB_RUN_DETAILS view, 3-115
 DBA_SCHEDULER_JOBS view, 3-115
 DBA_SCHEDULER_PROGRAM_ARGS view, 3-116
 DBA_SCHEDULER_PROGRAMS view, 3-116
 DBA_SCHEDULER_RUNNING_JOBS view, 3-116
 DBA_SCHEDULER_SCHEDULES view, 3-116
 DBA_SCHEDULER_WINDOW_DETAILS view, 3-116
 DBA_SCHEDULER_WINDOW_GROUPS view, 3-116
 DBA_SCHEDULER_WINDOW_LOG view, 3-116
 DBA_SCHEDULER_WINDOWS view, 3-117
 DBA_SCHEDULER_WINGROUP_MEMBERS view, 3-117
 DBA_SEC_RELEVANT_COLS view, 3-117
 DBA_SEGMENTS view, 3-117
 DBA_SEQUENCES view, 3-118
 DBA_SERVER_REGISTRY view, 3-118
 DBA_SERVICES view, 3-118
 DBA_SOURCE view, 3-119
 DBA_SOURCE_TABLES view, 3-119
 DBA_SQL_PROFILES view, 3-119
 DBA_SQLJ_TYPE_ATTRS view, 3-119
 DBA_SQLJ_TYPE_METHODS view, 3-119
 DBA_SQLJ_TYPES view, 3-120
 DBA_SQLSET view, 3-120
 DBA_SQLSET_BINDS view, 3-120
 DBA_SQLSET_REFERENCES view, 3-120
 DBA_SQLSET_STATEMENTS view, 3-121
 DBA_SQLTUNE_BINDS view, 3-122
 DBA_SQLTUNE_PLANS view, 3-122
 DBA_SQLTUNE_RATIONALE_PLAN view, 3-124
 DBA_SQLTUNE_STATISTICS view, 3-124
 DBA_STMT_AUDIT_OPTS view, 3-125
 DBA_STORED_SETTINGS view, 3-125
 DBA_STREAMS_ADMINISTRATOR view, 3-125
 DBA_STREAMS_GLOBAL_RULES view, 3-126
 DBA_STREAMS_MESSAGE_CONSUMERS view, 3-126
 DBA_STREAMS_MESSAGE_RULES view, 3-126
 DBA_STREAMS_NEWLY_SUPPORTED view, 3-126
 DBA_STREAMS_RULES view, 3-126
 DBA_STREAMS_SCHEMA_RULES view, 3-126
 DBA_STREAMS_TABLE_RULES view, 3-126
 DBA_STREAMS_TRANSFORM_FUNCTION view, 3-127
 DBA_STREAMS_UNSUPPORTED view, 3-127
 DBA_SUBPART_COL_STATISTICS view, 3-127
 DBA_SUBPART_HISTOGRAMS view, 3-127
 DBA_SUBPART_KEY_COLUMNS view, 3-127
 DBA_SUBPARTITION_TEMPLATES view, 3-127
 DBA_SUBSCRIBED_COLUMNS view, 3-127
 DBA_SUBSCRIBED_TABLES view, 3-128
 DBA_SUBSCRIPTIONS view, 3-128
 DBA_SYNONYMS view, 3-128
 DBA_SYS_PRIVS view, 3-128
 DBA_TAB_COL_STATISTICS view, 3-128
 DBA_TAB_COLS view, 3-128
 DBA_TAB_COLUMNS view, 3-129
 DBA_TAB_COMMENTS view, 3-129
 DBA_TAB_HISTOGRAMS view, 3-129
 DBA_TAB_MODIFICATIONS view, 3-129
 DBA_TAB_PARTITIONS view, 3-129
 DBA_TAB_PRIVS view, 3-129
 DBA_TAB_STATISTICS view, 3-130
 DBA_TAB_SUBPARTITIONS view, 3-130
 DBA_TABLES view, 3-130
 DBA_TABLESPACE_GROUPS view, 3-130
 DBA_TABLESPACES view, 3-130
 DBA_TEMP_FILES view, 3-132
 DBA_THRESHOLDS view, 3-132
 DBA_TRANSFORMATIONS view, 3-133
 DBA_TRIGGER_COLS view, 3-133
 DBA_TRIGGERS view, 3-133
 DBA_TS_QUOTAS view, 3-133
 DBA_TUNE_MVIEW view, 3-134
 DBA_TYPE_ATTRS view, 3-134
 DBA_TYPE_METHODS view, 3-134
 DBA_TYPE_VERSIONS view, 3-134
 DBA_TYPES view, 3-135
 DBA_UNDO_EXTENTS view, 3-135
 DBA_UNUSED_COL_TABS view, 3-135
 DBA_UPDATABLE_COLUMNS view, 3-135
 DBA_USERS view, 3-135
 DBA_USTATS view, 3-136
 DBA_VARRAYS view, 3-136
 DBA_VIEWS view, 3-136
 DBA_WAITERS view, 3-137
 DBA_WARNING_SETTINGS view, 3-137
 DBA_WORKSPACE_SESSIONS view, 2-3
 DBA_XML_SCHEMAS view, 3-137
 DBA_XML_TAB_COLS view, 3-137
 DBA_XML_TABLES view, 3-137
 DBA_XML_VIEW_COLS view, 3-137
 DBA_XML_VIEWS view, 3-139
 DBMS_ALERT_INFO view, 3-139
 DBMS_LOCK_ALLOCATED view, 3-139
 DBMSIOTC.SQL script, B-2
 DBMSOTRC.SQL script, B-2
 DBMSPOOL.SQL script, B-2
 DBWR_IO_SLAVES initialization parameter, 1-44
 DDL_WAIT_FOR_LOCKS initialization parameter, 1-45
 DEFCALL view, 2-3
 DEFCALLDEST view, 2-3
 DEFDEFAULTDEST view, 2-3
 DEFERRCOUNT view, 2-3
 DEFERROR view, 2-3
 DEFLOB view, 2-3

- DEFPROPAGATOR view, 2-3
- DEFSCCHEDULE view, 2-3
- DEFTRAN view, 2-3
- DEFTRANDEST view, 2-3
- DEPTREE view, 3-139
- destination
 - archiving redo log files, 1-70
 - overriding default, 1-70
 - USER_DUMP_DEST, 1-144
- DG_BROKER_CONFIG_FILE*n* initialization parameter, 1-45
- DG_BROKER_START initialization parameter, 1-46
- DICT synonym for DICTIONARY, 3-140
- DICTIONARY view, 3-140
 - DICT synonym, 3-140
- disk drives
 - archiving destination, 1-70
- DISK_ASYNC_IO initialization parameter, 1-46
- dispatcher processes
 - maximum number, 1-83
- DISPATCHERS initialization parameter, 1-47
- distinguished name, 1-115
- DISTRIBUTED_LOCK_TIMEOUT initialization parameter, 1-50
- DML_LOCKS initialization parameter, 1-50
- downgrading
 - scripts, B-4
- dump files, 1-84
- dynamic performance tables
 - CATALOG.SQL script, 4-1
 - public synonyms (V\$), 4-1
 - views (V_\$), 4-1

E

- embedded initialization parameter files, 1-60
- ENQUEUE_RESOURCES initialization parameter, 1-51
- enqueuees
 - AJV snapshot refresh, D-2
 - ALTER SYSTEM SET PARAMETER = VALUE, D-2
 - backup/restore, D-1
 - being written redo log, D-2
 - bind, D-1
 - buffer cache management, D-1
 - controlfile transaction, D-1
 - cross-instance call invocation, D-1
 - database mount, D-1
 - datafile, D-1
 - direct loader index creation, D-1
 - disabling, 1-50
 - distributed recovery process, D-1
 - distributed transaction, D-1
 - DML, D-2
 - ENQUEUE_RESOURCES initialization parameter, 1-51
 - extend table, D-2
 - file set, D-1

- high-water lock, D-1
- instance attribute lock, D-3
- instance number, D-1
- instance recovery, D-1
- instance registration lock, D-3
- instance state, D-1
- job queue, D-2
- library cache invalidation, D-1
- library cache lock, D-2
- library cache pin, D-2
- log start or switch, D-2
- media recovery, D-2
- mount definition, D-2
- object reuse, D-2
- parallel slave synchronization, D-2
- parallel slaves, D-2
- password file, D-2
- process startup, D-2
- redo log "kick", D-2
- redo thread, D-2
- row cache, D-2
- row wait, D-2
- sequence number, D-2
- sequence number value, D-2
- SMON, D-2
- sort segment, D-2
- space management transaction, D-2
- synchronized replication, D-2
- system commit number, D-2
- temporary segment, D-2
- temporary table, D-2
- temporary table object, D-2
- thread checkpoint, D-2
- transaction, D-2
- transaction recovery, D-2
- undo segment, serialization, D-2
- user name, D-2
- user-defined locks, D-2
- enterprise roles, 1-115
- ERROR_SIZE view, 3-140
- EVENT initialization parameter, 1-52
- EXCEPTIONS view, 3-140
- extent
 - allocating to instance, 1-62

F

- FAL_CLIENT initialization parameter, 1-52
- FAL_SERVER initialization parameter, 1-52
- FAST_START_IO_TARGET initialization parameter, 1-53
- FAST_START_MTTR_TARGET initialization parameter, 1-54
- FAST_START_PARALLEL_ROLLBACK initialization parameter, 1-54
- FILE_MAPPING initialization parameter, 1-55
- FILEIO_NETWORK_ADAPTERS initialization parameter, 1-55
- filenames
 - case significance, 1-4

- name and number, 1-56
- FILESYSTEMIO_OPTIONS initialization
 - parameter, 1-55
- FIXED_DATE initialization parameter, 1-56
- FLASHBACK_TRANSACTION_QUERY
 - view, 3-138
- free space list
 - locating space, 1-62
- FREELIST GROUPS clause, 1-62

G

- GC_FILES_TO_LOCKS initialization
 - parameter, 1-56
- GCS_SERVER_PROCESSES initialization
 - parameter, 1-57
- GLOBAL_CONTEXT_POOL_SIZE initialization
 - parameter, 1-58
- GLOBAL_NAME view, 3-138
- GLOBAL_NAMES initialization parameter, 1-58
- GV\$ views
 - for Real Application Clusters, 4-2

H

- HASH_AREA_SIZE initialization parameter, 1-58
- Heterogeneous Services
 - agents, 1-59
- HI_SHARED_MEMORY_ADDRESS initialization
 - parameter, 1-59
- HS_ALL_CAPS view, 3-138
- HS_ALL_DD view, 3-138
- HS_ALL_INITS view, 3-140
- HS_AUTOREGISTER initialization parameter, 1-59
- HS_BASE_CAPS view, 3-141
- HS_BASE_DD view, 3-141
- HS_CLASS_CAPS view, 3-141
- HS_CLASS_DD view, 3-141
- HS_CLASS_INIT view, 3-142
- HS_EXTERNAL_OBJECT_PRIVILEGES view, 3-142
- HS_EXTERNAL_OBJECTS view, 3-142
- HS_EXTERNAL_USER_PRIVILEGES view, 3-143
- HS_FDS_CLASS view, 3-143
- HS_FDS_INST view, 3-143
- HS_INST_CAPS view, 3-143
- HS_INST_DD view, 3-144
- HS_INST_INIT view, 3-144

I

- IDETREE view, 3-145
- IFILE initialization parameter, 1-5, 1-60
- IND synonym for USER_INDEXES view, 3-145
- INDEX_HISTOGRAM view, 3-145
- INDEX_STATS view, 3-145
- initialization parameter file
 - common file, 1-60
 - embedded, 1-60
 - INITDW.ORA
 - line continuation character, 1-5
 - overview, 1-3

- initialization parameters
 - ACTIVE_INSTANCE_COUNT, 1-17
 - altering, 1-6
 - AQ_TM_PROCESSES, 1-18
 - ARCHIVE_LAG_TARGET, 1-18
 - ASM_DISKGROUPS, 1-19
 - ASM_DISKSTRING, 1-20
 - ASM_POWER_LIMIT, 1-20
 - AUDIT_FILE_DEST, 1-21
 - AUDIT_SYS_OPERATIONS, 1-21
 - AUDIT_TRAIL, 1-21
 - BACKGROUND_CORE_DUMP, 1-22
 - BACKGROUND_DUMP_DEST, 1-22
 - BACKUP_TAPE_IO_SLAVES, 1-23
 - basic, 1-3
 - BITMAP_MERGE_AREA_SIZE, 1-24
 - BLANK_TRIMMING, 1-24
 - case significance in filenames, 1-4
 - CIRCUITS, 1-25
 - CLUSTER_DATABASE, 1-25
 - CLUSTER_DATABASE_INSTANCES, 1-25
 - CLUSTER_INTERCONNECTS, 1-26
 - COMMIT_POINT_STRENGTH, 1-27
 - COMPATIBLE, 1-27
 - CONTROL_FILE_RECORD_KEEP_TIME, 1-28
 - CONTROL_FILES, 1-28
 - CORE_DUMP_DEST, 1-29
 - CPU_COUNT, 1-29
 - CREATE_BITMAP_AREA_SIZE, 1-30
 - CREATE_STORED_OUTLINES, 1-30
 - CURSOR_SHARING, 1-31
 - CURSOR_SPACE_FOR_TIME, 1-32
 - DB_BLOCK_BUFFERS, 1-33
 - DB_BLOCK_CHECKING, 1-34
 - DB_BLOCK_CHECKSUM, 1-34
 - DB_BLOCK_SIZE, 1-35
 - DB_CACHE_ADVICE, 1-35
 - DB_CACHE_SIZE, 1-36
 - DB_CREATE_FILE_DEST, 1-37
 - DB_CREATE_ONLINE_LOG_DEST_n, 1-37
 - DB_DOMAIN, 1-38
 - DB_FILE_MULTIBLOCK_READ_COUNT, 1-38
 - DB_FILE_NAME_CONVERT, 1-39
 - DB_FILES, 1-40
 - DB_FLASHBACK_RETENTION_TARGET, 1-41
 - DB_KEEP_CACHE_SIZE, 1-41
 - DB_NAME, 1-41
 - DB_nK_CACHE_SIZE, 1-32
 - DB_RECOVERY_FILE_DEST, 1-42
 - DB_RECOVERY_FILE_DEST_SIZE, 1-43
 - DB_RECYCLE_CACHE_SIZE, 1-43
 - DB_UNIQUE_NAME, 1-43
 - DB_WRITER_PROCESSES, 1-44
 - DBWR_IO_SLAVES, 1-44
 - DDL_WAIT_FOR_LOCKS, 1-45
 - derived, 1-2
 - DG_BROKER_CONFIG_FILEn, 1-45
 - DG_BROKER_START, 1-46
 - DISK_ASYNC_IO, 1-46
 - DISPATCHERS, 1-47

display current settings, 1-15
DISTRIBUTED_LOCK_TIMEOUT, 1-50
DML_LOCKS, 1-50
embedded initialization parameter files, 1-5
ENQUEUE_RESOURCES, 1-51
EVENT, 1-52
FAL_CLIENT, 1-52
FAL_SERVER, 1-52
FAST_START_IO_TARGET, 1-53
FAST_START_MTTR_TARGET, 1-54
FAST_START_PARALLEL_ROLLBACK, 1-54
FILE_MAPPING, 1-55
FILEIO_NETWORK_ADAPTERS, 1-55
FILESYSTEMIO_OPTIONS, 1-55
FIXED_DATE, 1-56
functional category, 1-6
GC_FILES_TO_LOCKS, 1-56
GCS_SERVER_PROCESSES, 1-57
GLOBAL_CONTEXT_POOL_SIZE, 1-58
GLOBAL_NAMES, 1-58
HASH_AREA_SIZE, 1-58
HI_SHARED_MEMORY_ADDRESS, 1-59
HS_AUTOREGISTER, 1-59
IFILE, 1-5, 1-60
INSTANCE_GROUPS, 1-60
INSTANCE_NAME, 1-61
INSTANCE_NUMBER, 1-61
INSTANCE_TYPE, 1-62
JAVA_MAX_SESSIONSPACE_SIZE, 1-63
JAVA_POOL_SIZE, 1-63
JAVA_SOFT_SESSIONSPACE_LIMIT, 1-64
JOB_QUEUE_PROCESSES, 1-64
LARGE_POOL_SIZE, 1-65
LDAP_DIRECTORY_ACCESS, 1-65
LICENSE_MAX_SESSIONS, 1-66
LICENSE_MAX_USERS, 1-67
LICENSE_SESSIONS_WARNING, 1-67
LOCAL_LISTENER, 1-68
LOCK_NAME_SPACE, 1-68
LOCK_SGA, 1-69
LOG_ARCHIVE_CONFIG, 1-69
LOG_ARCHIVE_DEST, 1-70
LOG_ARCHIVE_DEST *n*, 1-71
LOG_ARCHIVE_DEST_STATE *n*, 1-72
LOG_ARCHIVE_DUPLEX_DEST, 1-73
LOG_ARCHIVE_FORMAT, 1-74
LOG_ARCHIVE_LOCAL_FIRST, 1-75
LOG_ARCHIVE_MAX_PROCESSES, 1-76
LOG_ARCHIVE_MIN_SUCCEED_DEST, 1-76
LOG_ARCHIVE_TRACE, 1-77
LOG_BUFFER, 1-78
LOG_CHECKPOINT_INTERVAL, 1-79
LOG_CHECKPOINT_TIMEOUT, 1-80
LOG_CHECKPOINTS_TO_ALERT, 1-81
LOG_FILE_NAME_CONVERT, 1-81
LOGMNR_MAX_PERSISTENT_SESSIONS, 1-82
MAX_COMMIT_PROPAGATION_DELAY, 1-82
MAX_DISPATCHERS, 1-83
MAX_DUMP_FILE_SIZE, 1-84
MAX_ENABLED_ROLES, 1-84
MAX_SHARED_SERVERS, 1-85
modifiable, 1-12
NLS_CALENDAR, 1-85
NLS_COMP, 1-86
NLS_CURRENCY, 1-86
NLS_DATE_FORMAT, 1-87
NLS_DATE_LANGUAGE, 1-87
NLS_DUAL_CURRENCY, 1-88
NLS_ISO_CURRENCY, 1-88
NLS_LANGUAGE, 1-89
NLS_LENGTH_SEMANTICS, 1-89
NLS_NCHAR_CONV_EXCP, 1-90
NLS_NUMERIC_CHARACTERS, 1-90
NLS_SORT, 1-91
NLS_TERRITORY, 1-91
NLS_TIMESTAMP_FORMAT, 1-92
NLS_TIMESTAMP_TZ_FORMAT, 1-92
O7_DICTIONARY_ACCESSIBILITY, 1-93
OBJECT_CACHE_MAX_SIZE_PERCENT, 1-93
OBJECT_CACHE_OPTIMAL_SIZE, 1-94
OLAP_PAGE_POOL_SIZE, 1-94
OPEN_CURSORS, 1-95
OPEN_LINKS, 1-95
OPEN_LINKS_PER_INSTANCE, 1-96
operating system-dependent values, 1-2
OPTIMIZER_DYNAMIC_SAMPLING, 1-96
OPTIMIZER_FEATURES_ENABLE, 1-97
OPTIMIZER_INDEX_CACHING, 1-99
OPTIMIZER_INDEX_COST_ADJ, 1-99
OPTIMIZER_MODE, 1-100
OS_AUTHENT_PREFIX, 1-100
OS_ROLES, 1-101
PARALLEL_ADAPTIVE_MULTI_USER, 1-102
PARALLEL_AUTOMATIC_TUNING, 1-102
PARALLEL_EXECUTION_MESSAGE_SIZE, 1-103
PARALLEL_INSTANCE_GROUP, 1-103
PARALLEL_MAX_SERVERS, 1-104
PARALLEL_MIN_PERCENT, 1-104
PARALLEL_MIN_SERVERS, 1-105
PARALLEL_THREADS_PER_CPU, 1-106
parameter files, 1-3
performance tuning, 1-2
PGA_AGGREGATE_TARGET, 1-106
PLSQL_CODE_TYPE, 1-107
PLSQL_COMPILER_FLAGS, 1-107
PLSQL_DEBUG, 1-108
PLSQL_NATIVE_LIBRARY_DIR, 1-109
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT, 1-109
PLSQL_OPTIMIZE_LEVEL, 1-109
PLSQL_V2_COMPATIBILITY, 1-110
PLSQL_WARNINGS, 1-111
PRE_PAGE_SGA, 1-112
PROCESSES, 1-112
QUERY_REWRITE_ENABLED, 1-113
QUERY_REWRITE_INTEGRITY, 1-114
RDBMS_SERVER_DN, 1-114
READ_ONLY_OPEN_DELAYED, 1-115
RECOVERY_PARALLELISM, 1-115

- REMOTE_ARCHIVE_ENABLE, 1-116
- REMOTE_DEPENDENCIES_MODE, 1-116
- REMOTE_LISTENER, 1-117
- REMOTE_LOGIN_PASSWORDFILE, 1-117
- REMOTE_OS_AUTHENT, 1-118
- REMOTE_OS_ROLES, 1-118
- REPLICATION_DEPENDENCY_TRACKING, 1-119
- RESOURCE_LIMIT, 1-119
- RESOURCE_MANAGER_PLAN, 1-120
- RESUMABLE_TIMEOUT, 1-120
- ROLLBACK_SEGMENTS, 1-121
- SERIAL_REUSE, 1-122
- SERVICE_NAMES, 1-123
- SESSION_CACHED_CURSORS, 1-123
- SESSION_MAX_OPEN_FILES, 1-124
- SESSIONS, 1-124
- SGA_MAX_SIZE, 1-125
- SGA_TARGET, 1-125
- SHADOW_CORE_DUMP, 1-126
- SHARED_MEMORY_ADDRESS, 1-127
- SHARED_POOL_RESERVED_SIZE, 1-127
- SHARED_POOL_SIZE, 1-127
- SHARED_SERVER_SESSIONS, 1-128
- SHARED_SERVERS, 1-129
- SKIP_UNUSABLE_INDEXES, 1-129
- SMTP_OUT_SERVER, 1-130
- SORT_AREA_RETAINED_SIZE, 1-130
- SORT_AREA_SIZE, 1-131
- specifying in initialization parameter files, 1-4
- SPFILE, 1-132
- SQL_TRACE, 1-132
- SQL92_SECURITY, 1-133
- SQLTUNE_CATEGORY, 1-133
- STANDBY_ARCHIVE_DEST, 1-134
- STANDBY_FILE_MANAGEMENT, 1-134
- STAR_TRANSFORMATION_ENABLED, 1-135
- STATISTICS_LEVEL, 1-135
- STREAMS_POOL_SIZE, 1-137
- TAPE_ASYNC_IO, 1-137
- THREAD, 1-137
- TIMED_OS_STATISTICS, 1-138
- TIMED_STATISTICS, 1-139
- TRACE_ENABLED, 1-139
- TRACEFILE_IDENTIFIER, 1-140
- TRANSACTIONS, 1-141
- TRANSACTIONS_PER_ROLLBACK_SEGMENT, 1-141
- UNDO_MANAGEMENT, 1-142
- UNDO_RETENTION, 1-142
- UNDO_TABLESPACE, 1-143
- USE_INDIRECT_DATA_BUFFERS, 1-143
- USER_DUMP_DEST, 1-144
- UTL_FILE_DIR, 1-144
 - variable, 1-2
- WORKAREA_SIZE_POLICY, 1-145
- INITJVM.SQL script, B-5
- INIT.ORA file. *See* initialization parameter file
- IN-list iterators
 - optimizer use of, 1-99

- inserts
 - locating free space, 1-62
- INST_ID column, 4-2
- INSTANCE clause
 - of ALTER TABLE
 - allocating extents, 1-62
- INSTANCE_GROUPS initialization parameter, 1-60
- INSTANCE_NAME initialization parameter, 1-61
- INSTANCE_NUMBER initialization parameter, 1-61
- INSTANCE_TYPE initialization parameter, 1-62
- instances
 - checkpoint, 1-79
 - startup order, 1-62
- I/O
 - reading multiple blocks, 1-39
 - redo log files, 1-78

J

- Java initialization parameters
 - JAVA_MAX_SESSIONSPACE_SIZE, 1-63
 - JAVA_POOL_SIZE, 1-63
 - JAVA_SOFT_SESSIONSPACE_LIMIT, 1-64
- Java scripts, B-5
- JAVA_MAX_SESSIONSPACE_SIZE initialization parameter, 1-63
- JAVA_POOL_SIZE initialization parameter, 1-63
- JAVA_SOFT_SESSIONSPACE_LIMIT initialization parameter, 1-64
- JOB_QUEUE_PROCESSES initialization parameter, 1-64

L

- language
 - database default language, 1-89
- LARGE_POOL_SIZE initialization parameter, 1-65
- LDAP_DIRECTORY_ACCESS initialization parameter, 1-65
- LICENSE_MAX_SESSIONS initialization parameter, 1-66
- LICENSE_MAX_USERS initialization parameter, 1-67
- LICENSE_SESSIONS_WARNING initialization parameter, 1-67
- licenses
 - maximum sessions, 1-66
 - maximum users, 1-67
 - warning for concurrent user sessions, 1-67
- limits on the database, A-1
- linguistic sorts, 1-91
- local currency, 1-87
- LOCAL_LISTENER initialization parameter, 1-68
- LOCK_NAME_SPACE initialization parameter, 1-68
- LOCK_SGA initialization parameter, 1-69
- locks
 - names, C-12
- LOG_ARCHIVE_CONFIG initialization parameter, 1-69
- LOG_ARCHIVE_DEST initialization

- parameter, 1-70
- LOG_ARCHIVE_DEST_1 initialization
 - parameter, 1-71
- LOG_ARCHIVE_DEST_STATE_1 initialization
 - parameter, 1-72
- LOG_ARCHIVE_DUPLEX_DEST initialization
 - parameter, 1-73
- LOG_ARCHIVE_FORMAT initialization
 - parameter, 1-74
- LOG_ARCHIVE_LOCAL_FIRST initialization
 - parameter, 1-75
- LOG_ARCHIVE_MAX_PROCESSES initialization
 - parameter, 1-76
- LOG_ARCHIVE_MIN_SUCCEED_DEST
 - initialization parameter, 1-76
- LOG_ARCHIVE_TRACE initialization
 - parameter, 1-77
- LOG_BUFFER initialization parameter, 1-78
- LOG_CHECKPOINT_INTERVAL initialization
 - parameter, 1-79
- LOG_CHECKPOINT_TIMEOUT initialization
 - parameter, 1-80
- LOG_CHECKPOINTS_TO_ALERT initialization
 - parameter, 1-81
- LOG_FILE_NAME_CONVERT initialization
 - parameter, 1-81
- LOGMNR_MAX_PERSISTENT_SESSIONS
 - initialization parameter, 1-82
- lowercase significance, 1-4

M

- MAP_OBJECT table, 3-146
- MAX_COMMIT_PROPAGATION_DELAY
 - initialization parameter, 1-82
- MAX_DISPATCHERS initialization parameter, 1-83
- MAX_DUMP_FILE_SIZE initialization
 - parameter, 1-84
- MAX_ENABLED_ROLES initialization
 - parameter, 1-84
- MAX_SHARED_SERVERS initialization
 - parameter, 1-85
- memory
 - amount used for sorting, 1-130
 - virtual, 1-2
- monitor
 - performance, 1-79
- multiple-process systems
 - number of processes, 1-113

N

- NLS_CALENDAR initialization parameter, 1-85
- NLS_COMP initialization parameter, 1-86
- NLS_CURRENCY initialization parameter, 1-86
 - defined by NLS_TERRITORY, 1-91
- NLS_DATABASE_PARAMETERS view, 3-147
- NLS_DATE_FORMAT initialization parameter, 1-87
 - defined by NLS_NUMERIC_
 - CHARACTERS, 1-90

- defined by NLS_TERRITORY, 1-91
- NLS_DATE_LANGUAGE initialization
 - parameter, 1-87
- NLS_DUAL_CURRENCY initialization
 - parameter, 1-88
- NLS_INSTANCE_PARAMETERS view, 3-147
- NLS_ISO_CURRENCY initialization
 - parameter, 1-88
 - defined by NLS_TERRITORY, 1-91
- NLS_LANGUAGE initialization parameter, 1-89
- NLS_LENGTH_SEMANTICS initialization
 - parameter, 1-89
- NLS_NCHAR_CONV_EXCP initialization
 - parameter, 1-90
- NLS_NUMERIC_CHARACTERS initialization
 - parameter, 1-90
- NLS_SESSION_PARAMETERS view, 3-147
- NLS_SORT initialization parameter, 1-91
- NLS_TERRITORY initialization parameter, 1-91
- NLS_TIMESTAMP_FORMAT initialization
 - parameter, 1-92
- NLS_TIMESTAMP_TZ_FORMAT initialization
 - parameter, 1-92
- numeric group separators, 1-90

O

- O7_DICTIONARY_ACCESSIBILITY initialization
 - parameter, 1-93
- OBJ synonym for USER_OBJECTS view, 3-147
- OBJECT_CACHE_MAX_SIZE_PERCENT
 - initialization parameter, 1-93
- OBJECT_CACHE_OPTIMAL_SIZE initialization
 - parameter, 1-94
- objects
 - data dictionary, 2-2
- OLAP_PAGE_POOL_SIZE initialization
 - parameter, 1-94
- online redo log
 - archiving mode, 1-70
 - block, 1-79
 - file size, 1-79
 - setting checkpoint interval, 1-79
- OPEN_CURSORS initialization parameter, 1-95
- OPEN_LINKS initialization parameter, 1-95
- OPEN_LINKS_PER_INSTANCE initialization
 - parameter, 1-96
- operating system
 - authenticating remote clients, 1-118
 - authenticating users and passwords, 1-100
 - dependent parameters, 1-2
 - roles for remote clients, 1-119
- optimization
 - trading cursor space for time, 1-32
- OPTIMIZER_DYNAMIC_SAMPLING initialization
 - parameter, 1-96
- OPTIMIZER_FEATURES_ENABLE initialization
 - parameter, 1-97
- OPTIMIZER_INDEX_CACHING initialization
 - parameter, 1-99

OPTIMIZER_INDEX_COST_ADJ initialization
parameter, 1-99
OPTIMIZER_MODE initialization parameter, 1-100
ORPHAN_KEY_TABLE view. *See* DBA_ORPHAN_
KEY_TABLE view.
OS_AUTHENT_PREFIX initialization
parameter, 1-100
OS_ROLES initialization parameter, 1-101

P

parallel query
maximum number of instances, 1-103
maximum number of servers, 1-104
minimum number of queries, 1-105
PARALLEL_ADAPTIVE_MULTI_USER initialization
parameter, 1-102
PARALLEL_AUTOMATIC_TUNING initialization
parameter, 1-102
PARALLEL_EXECUTION_MESSAGE_SIZE
initialization parameter, 1-103
PARALLEL_INSTANCE_GROUP initialization
parameter, 1-103
PARALLEL_MAX_SERVERS initialization
parameter, 1-104
PARALLEL_MIN_PERCENT initialization
parameter, 1-104
PARALLEL_MIN_SERVERS initialization
parameter, 1-105
PARALLEL_THREADS_PER_CPU initialization
parameter, 1-106
passwords
authenticating, 1-100, 1-118
REMOTE_LOGIN_PASSWORDFILE initialization
parameter, 1-118
PCM lock
mapping blocks to, 1-56
performance, 1-2
shared pool, 1-128
PGA_AGGREGATE_TARGET initialization
parameter, 1-106
PLAN_TABLE table, 3-147
PLSQL_CODE_TYPE initialization parameter, 1-107
PLSQL_COMPILER_FLAGS initialization
parameter, 1-107
PLSQL_DEBUG initialization parameter, 1-108
PLSQL_NATIVE_LIBRARY_DIR initialization
parameter, 1-109
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
initialization parameter, 1-109
PLSQL_OPTIMIZE_LEVEL initialization
parameter, 1-109
PLSQL_V2_COMPATIBILITY initialization
parameter, 1-110
PLSQL_WARNINGS initialization parameter, 1-111
PLUGGABLE_SET_CHECK view, 3-149
PRE_PAGE_SGA initialization parameter, 1-112
privileges
remote login, 1-118
RESTRICTED_SESSION privilege, 1-66, 1-67

table-level select privileges, 1-133
procedures
shared pool, 1-128
processes
derived parameter values, 1-141
dispatcher process maximum number, 1-83
maximum shared server processes, 1-85
number of server processes, 1-63, 1-64, 1-128,
1-129
recovery, 1-115
trace files, 1-144
user processes, 1-113
PROCESSES initialization parameter, 1-112
PRODUCT_COMPONENT_VERSION view, 3-149
profiles
resource limits, 1-119, 1-120
PROXY_USERS view, 3-150
PSTUBTBL view, 3-150
PUBLIC clause
of ALTER DATABASE
thread of redo, 1-138
PUBLIC_DEPENDENCY view, 3-150
PUBLICSYN view, 3-150

Q

queries
dictionary views, 1-56
hints, 1-100
QUERY_REWRITE_ENABLED initialization
parameter, 1-113
QUERY_REWRITE_INTEGRITY initialization
parameter, 1-114
QUEUE_PRIVILEGES view, 3-151

R

RC_ARCHIVED_LOG view, 2-4
RC_BACKUP_CONTROLFILE view, 2-4
RC_BACKUP_CORRUPTION view, 2-4
RC_BACKUP_DATAFILE view, 2-4
RC_BACKUP_PIECE view, 2-4
RC_BACKUP_REDOLOG view, 2-4
RC_BACKUP_SET view, 2-4
RC_BACKUP_SPFIL view, 2-4
RC_CHECKPOINT view, 2-4
RC_CONTROLFILE_COPY view, 2-4
RC_COPY_CORRUPTION view, 2-4
RC_DATABASE view, 2-4
RC_DATABASE_BLOCK_CORRUPTION view, 2-4
RC_DATABASE_INCARNATION view, 2-4
RC_DATAFILE view, 2-4
RC_DATAFILE_COPY view, 2-4
RC_LOG_HISTORY view, 2-4
RC_OFFLINE_RANGE view, 2-4
RC_PROXY_CONTROLFILE view, 2-4
RC_PROXY_DATAFILE view, 2-4
RC_REDO_LOG view, 2-4
RC_REDO_THREAD view, 2-4
RC_RESYNC view, 2-4

RC_RMAN_CONFIGURATION view, 2-4
 RC_STORED_SCRIPT view, 2-4
 RC_STORED_SCRIPT_LINE view, 2-4
 RC_TABLESPACE view, 2-4
 RDBMS_SERVER_DN initialization
 parameter, 1-114
 READ_ONLY_OPEN_DELAYED initialization
 parameter, 1-115
 Real Application Clusters
 dynamic performance views, 4-2
 initialization parameters
 CIRCUITS, 1-25
 CLUSTER_DATABASE, 1-25
 CLUSTER_DATABASE_INSTANCES, 1-25
 CLUSTER_INTERCONNECTS, 1-26
 DISPATCHERS, 1-47
 FAST_START_PARALLEL_ROLLBACK, 1-54
 PARALLEL_ADAPTIVE_MULTI_
 USER, 1-102
 PARALLEL_AUTOMATIC_TUNING, 1-102
 PARALLEL_EXECUTION_MESSAGE_
 SIZE, 1-103
 PARALLEL_INSTANCE_GROUP, 1-103
 PARALLEL_MAX_SERVERS, 1-104
 PARALLEL_MIN_PERCENT, 1-104
 PARALLEL_MIN_SERVERS, 1-105
 PARALLEL_THREADS_PER_CPU, 1-106
 recovery
 number of required processes, 1-115
 RECOVERY_PARALLELISM initialization
 parameter, 1-115
 RECYCLEBIN synonym for USER_RECYCLEBIN
 view, 3-151
 redo logs
 buffer size, 1-78
 redo thread, 1-138
 remote clients
 authenticating, 1-118
 remote logins, 1-118
 REMOTE_ARCHIVE_ENABLE initialization
 parameter, 1-116
 REMOTE_DEPENDENCIES_MODE initialization
 parameter, 1-116
 REMOTE_LISTENER initialization parameter, 1-117
 REMOTE_LOGIN_PASSWORDFILE initialization
 parameter, 1-117
 REMOTE_OS_AUTHENT initialization
 parameter, 1-118
 REMOTE_OS_ROLES initialization parameter, 1-118
 REPAIR_TABLE view. *See* DBA_REPAIR_TABLE
 view.
 REPLICATION_DEPENDENCY_TRACKING
 initialization parameter, 1-119
 resource limits for profiles, 1-119, 1-120
 RESOURCE_COST view, 3-151
 RESOURCE_LIMIT initialization parameter, 1-119
 RESOURCE_MANAGER_PLAN initialization
 parameter, 1-120
 RESOURCE_MAP table, 3-151
 RESUMABLE_TIMEOUT initialization
 parameter, 1-120
 RMJVM.SQL script, B-5
 ROLE_ROLE_PRIVS view, 3-151
 ROLE_SYS_PRIVS view, 3-152
 ROLE_TAB_PRIVS view, 3-152
 ROLE_WM_PRIVS view, 2-3
 roles, 1-101
 remote clients, 1-119
 ROLLBACK_SEGMENTS initialization
 parameter, 1-121

S

scripts
 CATALOG.SQL, B-1
 CATBLOCK.SQL, B-2
 CATCLUST.SQL, B-1
 CATEXP7.SQL, B-2
 CATHS.SQL, B-2
 CATIO.SQL, B-2
 CATJAVA.SQL, B-5
 CATNOADT.SQL, B-4
 CATNOAUD.SQL, B-4
 CATNOHS.SQL, B-4
 CATNOPRT.SQL, B-4
 CATNOQUE.SQL, B-4
 CATNORMN.SQL, B-4
 CATNOSVM.SQL, B-4
 CATNSNMP.SQL, B-4
 CATOCTK.SQL, B-2
 CATPROC.SQL, B-1
 CATQUEUE.SQL, B-2
 CATREP.SQL, B-2
 CATRMAN.SQL, B-2
 D0902000.SQL, B-5
 data dictionary, B-2
 DBMSIOTC.SQL, B-2
 DBMSOTRC.SQL, B-2
 DBMSPOOL.SQL, B-2
 downgrading, B-4
 INITJVM.SQL, B-5
 Java, B-5
 RMJVM.SQL, B-5
 SQL scripts, B-1
 U0800060.SQL, B-5
 U0801070.SQL, B-5
 U0900010.SQL, B-5
 U0902000.SQL, B-5
 upgrading, B-4
 USERLOCK.SQL, B-2
 UTLBSTAT.SQL, B-3
 UTLCHN1.SQL, B-3
 UTLCONST.SQL, B-3
 UTLDTREE.SQL, B-3
 UTLESTAT.SQL, B-3
 UTLEXPT1.SQL, B-3
 UTLIP.SQL, B-3
 UTLIRP.SQL, B-3
 UTLLOCKT.SQL, B-3
 UTLPWDMG.SQL, B-3

- UTLRP.SQL, B-3
- UTLSAMPL.SQL, B-3
- UTLSCLN.SQL, B-3
- UTLTKPRF.SQL, B-4
- UTLU101I.SQL, B-5
- UTLU101S.SQL, B-5
- UTLVALID.SQL, B-4
- UTLXPLAN.SQL, B-4
- SEQ synonym for USER_SEQUENCES view, 3-152
- SERIAL_REUSE initialization parameter, 1-122
- server parameter file
 - overview, 1-3
- server processes
 - number, 1-63, 1-64, 1-128, 1-129
- SERVICE_NAMES initialization parameter, 1-123
- SESSION_CACHED_CURSORS initialization parameter, 1-123
- SESSION_CONTEXT view, 3-152
- SESSION_MAX_OPEN_FILES initialization parameter, 1-124
- SESSION_PRIVS view, 3-152
- SESSION_ROLES view, 3-153
- SESSIONS initialization parameter, 1-124
- SGA_MAX_SIZE initialization parameter, 1-125
- SGA_TARGET initialization parameter, 1-125
- SHADOW_CORE_DUMP initialization parameter, 1-126
- shared server
 - SESSIONS initialization parameter, 1-124
- shared server processes
 - maximum number, 1-85
- SHARED_MEMORY_ADDRESS initialization parameter, 1-127
- SHARED_POOL_RESERVED_SIZE initialization parameter, 1-127
- SHARED_POOL_SIZE initialization parameter, 1-127
- SHARED_SERVER_SESSIONS initialization parameter, 1-128
- SHARED_SERVERS initialization parameter, 1-129
- SHOW PARAMETERS command, 1-15
- SKIP_UNUSABLE_INDEXES initialization parameter, 1-129
- SMTP_OUT_SERVER initialization parameter, 1-130
- SORT_AREA_RETAINED_SIZE initialization parameter, 1-130
- SORT_AREA_SIZE initialization parameter, 1-131
- sorting
 - maximum memory, 1-130
 - ORDER BY queries, 1-91
- SOURCE_SIZE view, 3-153
- SPFILE initialization parameter, 1-132
- SQL scripts
 - creating the data dictionary, B-1
 - optional, B-2
 - required, B-1
- SQL_TRACE initialization parameter, 1-132
- SQL92_SECURITY initialization parameter, 1-133
- SQLTUNE_CATEGORY initialization parameter, 1-133
- STANDBY_ARCHIVE_DEST initialization parameter, 1-134
- STANDBY_FILE_MANAGEMENT initialization parameter, 1-134
- STAR_TRANSFORMATION_ENABLED initialization parameter, 1-135
- starting up
 - instance number, 1-62
 - startup order, 1-62
- statistics
 - background checkpoints completed, E-2
 - background checkpoints started, E-2
 - background timeouts, E-2
 - branch node splits, E-2
 - buffer is not pinned count, E-2
 - buffer is pinned count, E-2
 - bytes received via SQL*Net from client, E-2
 - bytes received via SQL*Net from dblink, E-2
 - bytes sent via SQL*Net to client, E-2
 - bytes sent via SQL*Net to dblink, E-2
 - Cached Commit SCN referenced, E-2
 - calls to get snapshot scn
 - kcmgss, E-2
 - calls to kcmgas, E-3
 - calls to kcmgcs, E-3
 - calls to kcmgrs, E-3
 - change write time, E-3
 - cleanouts and rollbacks - consistent read gets, E-3
 - cleanouts only - consistent read gets, E-3
 - cluster key scan block gets, E-3
 - cluster key scans, E-3
 - cold recycle reads, E-3
 - commit cleanout failures
 - block lost, E-3
 - buffer being written, E-3
 - callback failure, E-3
 - cannot pin, E-3
 - hot backup in progress, E-3
 - write disabled, E-3
 - commit cleanouts, E-3
 - commit cleanouts successfully completed, E-3
 - Commit SCN cached, E-3
 - consistent changes, E-3
 - consistent gets, E-4
 - CPU used by this session, E-4
 - CPU used when call started, E-4
 - CR blocks created, E-4
 - current blocks converted for CR, E-4
 - cursor authentication, E-4
 - data blocks consistent reads - undo records
 - applied, E-4
 - db block changes, E-4
 - db block gets, E-4
 - DBWR buffers scanned, E-4
 - DBWR checkpoint buffers written, E-4
 - DBWR checkpoints, 1-79, E-4
 - DBWR cross instance writes, E-4
 - DBWR free buffers found, E-4
 - DBWR lru scans, E-5
 - DBWR make free requests, E-5

- DBWR revisited being-written buffer, E-5
- DBWR summed scan depth, E-5
- DBWR transaction table writes, E-5
- DBWR undo block writes, E-5
- DDL statements parallelized, E-5
- deferred (CURRENT) block cleanout applications, E-5
- DFO trees parallelized, E-5
- dirty buffers inspected, E-5
- DML statements parallelized, E-5
- enqueue conversions, E-5
- enqueue deadlocks, E-5
- enqueue releases, E-5
- enqueue requests, E-5
- enqueue timeouts, E-5
- enqueue waits, E-5
- exchange deadlocks, E-5
- execute count, E-6
- free buffer inspected, E-6
- free buffer requested, E-6
- global cache blocks corrupt, E-6
- global cache cr block log flush time, E-6, E-7
- global cache cr block send time, E-6
- global cache cr timeouts, E-6
- global cache defers, E-6
- global cache freelist waits, E-6
- global cache get time, E-6
- global cache gets, E-6
- global cache prepare failures, E-7
- global lock async converts, E-7
- global lock async gets, E-7
- global lock convert time, E-7
- global lock get time, E-7
- global lock release time, E-7
- global lock releases, E-7
- global lock sync converts, E-7
- global lock sync gets, E-7
- hot buffers moved to head of LRU, E-7
- immediate (CR) block cleanout applications, E-7
- immediate (CURRENT) block cleanout applications, E-7
- index fast full scans (direct read), E-7
- index fast full scans (full), E-7
- index fast full scans (rowid ranges), E-7
- instance recovery database freeze count, E-7
- kcmccs called get current scn, E-7
- kcmccs read scn without going to DLM, E-7
- kcmggs waited for batching, E-7
- leaf node splits, E-7
- logons cumulative, E-7
- logons current, E-7
- messages received, E-7
- messages sent, E-8
- native hash arithmetic execute, E-8
- native hash arithmetic fail, E-8
- next scns gotten without going to DLM, E-8
- no buffer to keep pinned count, E-8
- no work - consistent read gets, E-8
- opened cursors cumulative, E-8
- opened cursors current, E-8
- opens of replaced files, E-8
- opens requiring cache replacement, E-8
- operating system
 - OS All other sleep time, E-8
 - OS Chars read and written, E-8
 - OS Data page fault sleep time, E-8
 - OS Input blocks, E-8
 - OS Involuntary context switches, E-8
 - OS Kernel page fault sleep time, E-8
 - OS Major page faults, E-8
 - OS Messages received, E-8
 - OS Messages sent, E-8
 - OS Minor page faults, E-8
 - OS Other system trap CPU time, E-8
 - OS Output blocks, E-9
 - OS Process heap time, E-9
 - OS Signals received, E-9
 - OS Swaps, E-9
 - OS System call CPU time, E-9
 - OS System calls, E-9
 - OS Text page fault sleep time, E-9
 - OS User level CPU time, E-9
 - OS User lock wait sleep time, E-9
 - OS Voluntary context switches, E-9
 - OS Wait-cpu (latency) time, E-9
 - Process stack size, E-9
- Parallel operations downgraded 1 to 25 pct, E-9
- Parallel operations downgraded 25 to 50 pct, E-9
- Parallel operations downgraded 50 to 75 pct, E-9
- Parallel operations downgraded 75 to 99 pct, E-9
- Parallel operations downgraded to serial, E-9
- Parallel operations not downgraded, E-9
- parse count (hard), E-9
- parse count (total), E-9
- parse time cpu, E-9
- parse time elapsed, E-10
- physical reads, E-10
- physical reads direct, E-10
- physical writes, E-10
- physical writes direct, E-10
- physical writes non checkpoint, E-10
- pinned buffers inspected, E-10
- prefetched blocks, E-10
- prefetched blocks aged out before use, E-10
- process last non-idle time, E-10
- PX local messages recv'd, E-10
- PX local messages sent, E-10
- PX remote messages recv'd, E-10
- PX remote messages sent, E-10
- queries parallelized, E-10
- recovery array read time, E-10
- recovery array reads, E-10
- recovery blocks read, E-10
- recursive calls, E-11
- recursive cpu usage, E-11
- redo blocks written, E-11
- redo buffer allocation retries, E-11
- redo entries, E-11
- redo log space requests, E-11
- redo log space wait time, E-11

- redo log switch interrupts, E-11
- redo ordering marks, E-11
- redo size, E-11
- redo sync writes, E-11
- redo synch time, E-11
- redo wastage, E-12
- redo write time, E-12
- redo writer latching time, E-12
- redo writes, E-12
- remote instance undo block writes, E-12
- remote instance undo header writes, E-12
- rollback changes - undo records applied, E-12
- rollbacks only - consistent read gets, E-12
- rows fetched via callback, E-12
- serializable aborts, E-12
- session connect time, E-12
- session cursor cache count, E-12
- session cursor cache hits, E-12
- session logical reads, E-12
- session pga memory, E-12
- session pga memory max, E-12
- session stored procedure space, E-12
- session uga memory, E-12
- session uga memory max, E-12
- sorts (disk), E-13
- sorts (memory), E-13
- sorts (rows), E-13
- SQL*Net roundtrips to/from client, E-13
- SQL*Net roundtrips to/from dblink, E-13
- summed dirty queue length, E-13
- switch current to new buffer, E-13
- table fetch by rowid, E-13
- table fetch continued row, E-13
- table scan blocks gotten, E-13
- table scan rows gotten, E-13
- table scans (cache partitions), E-13
- table scans (direct read), E-14
- table scans (long tables), E-14
- table scans (rowid ranges), E-14
- table scans (short tables), E-14
- timed, 1-139
- total file opens, E-14
- transaction lock background get time, E-14
- transaction lock background gets, E-14
- transaction lock foreground requests, E-14
- transaction lock foreground wait time, E-14
- transaction rollbacks, E-14
- transaction tables consistent read rollbacks, E-14
- transaction tables consistent reads - undo records applied, E-14
- Unnecessary process cleanup for SCN
 - batching, E-14
- user calls, E-14
- user commits, E-14
- user rollbacks, E-15
- write clones created in background, E-15
- write clones created in foreground, E-15
- STATISTICS_LEVEL initialization parameter, 1-135
- STMT_AUDIT_OPTION_MAP table, 3-153
- STREAMS_POOL_SIZE initialization

- parameter, 1-137
- switch redo log file, 1-79
- SYN synonym for USER_SYNONYMS view, 3-153
- SYNONYMS view, 3-153
- SYS_OBJECTS view, 3-154
- SYSCATALOG view, 3-153
- SYSFILES view, 3-153
- SYSSEGOBJ view, 3-154
- system global area, 1-2
 - buffer areas, 1-78
 - maximum concurrent transactions, 1-141
- system performance, 1-2
- SYSTEM_PRIVILEGE_MAP table, 3-154

T

- TAB view, 3-154
- TABLE_PRIVILEGE_MAP table, 3-154
- tables
 - data dictionary, 2-2
 - DML locks, 1-50
 - locating free space, 1-62
- TABQUOTAS view, 3-155
- TABS synonym for USER_TABLES view, 3-154
- tape archiving destination, 1-70
- TAPE_ASYNCH_IO initialization parameter, 1-137
- territory, 1-88, 1-91
- THREAD initialization parameter, 1-137
- TIMED_OS_STATISTICS initialization
 - parameter, 1-138
- TIMED_STATISTICS initialization parameter, 1-139
 - file read/write statistics, 4-61, 5-87
- TO_CHAR function, 1-87
- TO_DATE function, 1-87
- trace files
 - destination, 1-144
 - maximum size, 1-84
- TRACE_ENABLED initialization parameter, 1-139
- TRACEFILE_IDENTIFIER initialization
 - parameter, 1-140
- transactions
 - concurrent, 1-141
 - data locking, 1-50
- TRANSACTIONS initialization parameter, 1-141
 - DML locks, 1-50
- TRANSACTIONS_PER_ROLLBACK_SEGMENT
 - initialization parameter, 1-141
- TRUSTED_SERVERS view, 3-155
- TS_PITR_CHECK view, 3-155
- TS_PITR_OBJECTS TO_BE_DROPPED view, 3-156

U

- U0800060.SQL script, B-5
- U0801070.SQL script, B-5
- U0900010.SQL script, B-5
- U0902000.SQL script, B-5
- UNDO_MANAGEMENT initialization
 - parameter, 1-142
- UNDO_RETENTION initialization parameter, 1-142

UNDO_TABLESPACE initialization
 parameter, 1-143
 UNI_PLUGGABLE_SET_CHECK view, 3-156
 updates
 locating free space, 1-62
 upgrading
 scripts, B-4
 uppercase characters, 1-4
 USE_INDIRECT_DATA_BUFFERS initialization
 parameter, 1-143
 user processes
 trace files, 1-144
 USER_ADVISOR_ACTIONS view, 3-157
 USER_ADVISOR_FINDINGS view, 3-157
 USER_ADVISOR_JOURNAL view, 3-157
 USER_ADVISOR_LOG view, 3-157
 USER_ADVISOR_OBJECTS view, 3-157
 USER_ADVISOR_PARAMETERS view, 3-157
 USER_ADVISOR_RATIONALE view, 3-157
 USER_ADVISOR_RECOMMENDATIONS
 view, 3-158
 USER_ADVISOR_SQLA_REC_SUM view, 3-158
 USER_ADVISOR_SQLA_WK_MAP view, 3-158
 USER_ADVISOR_SQLA_WK_STMTS view, 3-158
 USER_ADVISOR_SQLW_JOURNAL view, 3-158
 USER_ADVISOR_SQLW_PARAMETERS
 view, 3-158
 USER_ADVISOR_SQLW_STMTS view, 3-158
 USER_ADVISOR_SQLW_SUM view, 3-159
 USER_ADVISOR_SQLW_TABLES view, 3-159
 USER_ADVISOR_SQLW_TEMPLATES view, 3-159
 USER_ADVISOR_TASKS view, 3-159
 USER_ADVISOR_TEMPLATES view, 3-159
 USER_ALL_TABLES view, 3-159
 USER_AQ_AGENT_PRIVS view, 3-159
 USER_ARGUMENTS view, 3-160
 USER_ASSOCIATIONS view, 3-160
 USER_ATTRIBUTE_TRANSFORMATIONS
 view, 3-160
 USER_AUDIT_OBJECT view, 3-160
 USER_AUDIT_POLICIES view, 3-160
 USER_AUDIT_POLICY_COLUMNS view, 3-160
 USER_AUDIT_SESSION view, 3-160
 USER_AUDIT_STATEMENT view, 3-161
 USER_AUDIT_TRAIL view, 3-161
 USER_AW_PS view, 3-161
 USER_AWS view, 3-161
 USER_BASE_TABLE_MVIEWS view, 3-161
 USER_CATALOG view, 3-161
 CAT synonym, 2-145
 USER_CLU_COLUMNS view, 3-161
 USER_CLUSTER_HASH_EXPRESSIONS
 view, 3-161
 USER_CLUSTERS view, 3-161
 CLU synonym, 2-147
 USER_COL_COMMENTS view, 3-162
 USER_COL_PRIVS view, 3-162
 USER_COL_PRIVS_MADE view, 3-162
 USER_COL_PRIVS_RECD view, 3-162
 USER_COLL_TYPES view, 3-162
 USER_CONS_COLUMNS view, 3-162
 USER_CONS_OBJ_COLUMNS view, 3-162
 USER_CONSTRAINTS view, 3-163
 USER_DATAPUMP_JOBS view, 3-163
 USER_DB_LINKS view, 3-163
 USER_DEPENDENCIES view, 3-163
 USER_DIM_ATTRIBUTES view, 3-163
 USER_DIM_CHILD_OF view, 3-163
 USER_DIM_HIERARCHIES view, 3-163
 USER_DIM_JOIN_KEY view, 3-163
 USER_DIM_LEVEL_KEY view, 3-164
 USER_DIM_LEVELS view, 3-164
 USER_DIMENSIONS view, 3-164
 USER_DUMP_DEST initialization parameter, 1-144
 USER_ERRORS view, 3-164
 USER_EVALUATION_CONTEXT_TABLES
 view, 3-164
 USER_EVALUATION_CONTEXT_VARS
 view, 3-164
 USER_EVALUATION_CONTEXTS view, 3-164
 USER_EXTENTS view, 3-164
 USER_EXTERNAL_LOCATIONS view, 3-165
 USER_EXTERNAL_TABLES view, 3-165
 USER_FREE_SPACE view, 3-165
 USER_IND_COLUMNS view, 3-165
 USER_IND_EXPRESSIONS view, 3-165
 USER_IND_PARTITIONS view, 3-165
 USER_IND_STATISTICS view, 3-165
 USER_IND_SUBPARTITIONS view, 3-165
 USER_INDEXES view, 3-166
 IND synonym, 3-145
 USER_INDEXTYPE_ARRAYTYPES view, 3-166
 USER_INDEXTYPE_COMMENTS view, 3-166
 USER_INDEXTYPE_OPERATORS view, 3-166
 USER_INDEXTYPES view, 3-166
 USER_INTERNAL_TRIGGERS view, 3-166
 USER_JAVA_ARGUMENTS view, 3-166
 USER_JAVA_CLASSES view, 3-167
 USER_JAVA_DERIVATIONS view, 3-167
 USER_JAVA_FIELDS view, 3-167
 USER_JAVA_IMPLEMENTATIONS view, 3-167
 USER_JAVA_INNERS view, 3-167
 USER_JAVA_LAYOUTS view, 3-167
 USER_JAVA_METHODS view, 3-167
 USER_JAVA_NCOMPS view, 3-168
 USER_JAVA_POLICY view, 3-168
 USER_JAVA_RESOLVERS view, 3-168
 USER_JAVA_THROWS view, 3-168
 USER_JOBS view, 3-168
 USER_JOIN_IND_COLUMNS view, 3-168
 USER_LIBRARIES view, 3-168
 USER_LOB_PARTITIONS view, 3-168
 USER_LOB_SUBPARTITIONS view, 3-169
 USER_LOB_TEMPLATES view, 3-169
 USER_LOBS view, 3-169
 USER_LOG_GROUP_COLUMNS view, 3-169
 USER_LOG_GROUPS view, 3-169
 USER_METHOD_PARAMS view, 3-169
 USER_METHOD_RESULTS view, 3-169
 USER_MVIEW_AGGREGATES view, 3-169

USER_MVIEW_ANALYSIS view, 3-170
 USER_MVIEW_COMMENTS view, 3-170
 USER_MVIEW_DETAIL_RELATIONS view, 3-170
 USER_MVIEW_JOINS view, 3-170
 USER_MVIEW_KEYS view, 3-170
 USER_MVIEW_LOGS view, 3-170
 USER_MVIEW_REFRESH_TIMES view, 3-170
 USER_MVIEWS view, 3-171
 USER_NESTED_TABLE_COLS view, 3-171
 USER_NESTED_TABLES view, 3-171
 USER_OBJ_AUDIT_OPTS view, 3-171
 USER_OBJ_COLATTRS view, 3-171
 USER_OBJECT_SIZE view, 3-171
 USER_OBJECT_TABLES view, 3-171
 USER_OBJECTS view, 3-171
 OBJ synonym, 3-147
 USER_OPANCILLARY view, 3-172
 USER_OPARGUMENTS view, 3-172
 USER_OPBINDINGS view, 3-172
 USER_OPERATOR_COMMENTS view, 3-172
 USER_OPERATORS view, 3-172
 USER_OUTLINE_HINTS view, 3-172
 USER_OUTLINES view, 3-172
 USER_PART_COL_STATISTICS view, 3-172
 USER_PART_HISTOGRAMS view, 3-173
 USER_PART_INDEXES view, 3-173
 USER_PART_KEY_COLUMNS view, 3-173
 USER_PART_LOBS view, 3-173
 USER_PART_TABLES view, 3-173
 USER_PARTIAL_DROP_TABS view, 3-173
 USER_PASSWORD_LIMITS view, 3-173
 USER_PENDING_CONV_TABLES view, 3-174
 USER_PLSQL_OBJECT_SETTINGS view, 3-174
 USER_POLICIES view, 3-174
 USER_POLICY_CONTEXTS view, 3-174
 USER_POLICY_GROUPS view, 3-174
 USER_PROCEDURES view, 3-174
 USER_PROXIES view, 3-174
 USER_PUBLISHED_COLUMNS view, 3-175
 USER_QUEUE_SCHEDULES view, 3-175
 USER_QUEUE_TABLES view, 3-175
 USER_QUEUES view, 3-175
 USER_RECYCLEBIN view, 3-175
 RECYCLEBIN synonym, 3-151
 USER_REFRESH view, 3-175
 USER_REFRESH_CHILDREN view, 3-175
 USER_REFS view, 3-175
 USER_REGISTERED_MVIEWS view, 3-176
 USER_REGISTRY view, 3-176
 USER_REPCAT view, 2-2
 USER_REPCAT_REFRESH_TEMPLATES view, 2-2
 USER_REPCAT_TEMPLATE_OBJECTS view, 2-2
 USER_REPCAT_TEMPLATE_PARAMS view, 2-2
 USER_REPCAT_TEMPLATE_SITES view, 2-2
 USER_REPCAT_USER_AUTHORIZATION
 view, 2-2
 USER_REPCAT_USER_PARM_VALUES view, 2-2
 USER_REPCATLOG view, 2-2
 USER_REPCOLUMN view, 2-2
 USER_REPCOLUMN_GROUP view, 2-2
 USER_REPCONFLICT view, 2-2
 USER_REPDDL view, 2-2
 USER_REPFLAVOR_COLUMNS view, 2-2
 USER_REPFLAVOR_OBJECTS view, 2-2
 USER_REPFLAVORS view, 2-2
 USER_REPGENERATED view, 2-2
 USER_REPGENOBJECTS view, 2-2
 USER_REPGROUP view, 2-2
 USER_REPGROUP_PRIVILEGES view, 2-2
 USER_REPGROUPED_COLUMN view, 2-2
 USER_REPKEY_COLUMNS view, 2-2
 USER_REPOBJECT view, 2-2
 USER_REPPARAMETER_COLUMN view, 2-2
 USER_REPPRIORITY view, 2-2
 USER_REPPRIORITY_GROUP view, 2-2
 USER_REPPROP view, 2-2
 USER_REPRESOL_STATS_CONTROL view, 2-2
 USER_REPRESOLUTION view, 2-2
 USER_REPRESOLUTION_METHOD view, 2-2
 USER_REPRESOLUTION_STATISTICS view, 2-2
 USER_REPSHEMA view, 2-3
 USER_REPSITES view, 2-3
 USER_RESOURCE_LIMITS view, 3-176
 USER_RESUMABLE view, 3-176
 USER_REWRITE_EQUIVALENCES view, 3-176
 USER_ROLE_PRIVS view, 3-176
 USER_RSRC_CONSUMER_GROUP_PRIVS
 view, 3-177
 USER_RSRC_MANAGER_SYSTEM_PRIVS
 view, 3-177
 USER_RULE_SET_RULES view, 3-177
 USER_RULE_SETS view, 3-177
 USER_RULES view, 3-177
 USER_SCHEDULER_JOB_ARGS view, 3-177
 USER_SCHEDULER_JOB_LOG view, 3-177
 USER_SCHEDULER_JOB_RUN_DETAILS
 view, 3-178
 USER_SCHEDULER_JOBS view, 3-178
 USER_SCHEDULER_PROGRAM_ARGS
 view, 3-178
 USER_SCHEDULER_PROGRAMS view, 3-178
 USER_SCHEDULER_RUNNING_JOBS view, 3-178
 USER_SCHEDULER_SCHEDULES view, 3-178
 USER_SEC_RELEVANT_COLS view, 3-178
 USER_SEGMENTS view, 3-179
 USER_SEQUENCES view, 3-179
 SEQ synonym, 3-152
 USER_SOURCE view, 3-179
 USER_SOURCE_TABLES view, 3-179
 USER_SQLJ_TYPE_ATTRS view, 3-179
 USER_SQLJ_TYPE_METHODS view, 3-179
 USER_SQLJ_TYPES view, 3-179
 USER_SQLSET view, 3-179
 USER_SQLSET_BINDS view, 3-180
 USER_SQLSET_REFERENCES view, 3-180
 USER_SQLSET_STATEMENTS view, 3-180
 USER_SQLTUNE_BINDS view, 3-180
 USER_SQLTUNE_PLANS view, 3-180
 USER_SQLTUNE_RATIONALE_PLAN view, 3-180
 USER_SQLTUNE_STATISTICS view, 3-180

USER_STORED_SETTINGS view, 3-181
 USER_SUBPART_COL_STATISTICS view, 3-181
 USER_SUBPART_HISTOGRAMS view, 3-181
 USER_SUBPART_KEY_COLUMNS view, 3-181
 USER_SUBPARTITION_TEMPLATES view, 3-181
 USER_SUBSCRIBED_COLUMNS view, 3-181
 USER_SUBSCRIBED_TABLES view, 3-181
 USER_SUBSCRIPTIONS view, 3-182
 USER_SYNONYMS view, 3-182
 SYN synonym, 3-153
 USER_SYS_PRIVS view, 3-182
 USER_TAB_COL_STATISTICS view, 3-182
 USER_TAB_COLS view, 3-182
 USER_TAB_COLUMNS view, 3-182
 COLS synonym, 2-147
 USER_TAB_COMMENTS view, 3-182
 USER_TAB_HISTOGRAMS view, 3-182
 USER_TAB_MODIFICATIONS view, 3-183
 USER_TAB_PARTITIONS view, 3-183
 USER_TAB_PRIVS view, 3-183
 USER_TAB_PRIVS_MADE view, 3-183
 USER_TAB_PRIVS_RECD view, 3-183
 USER_TAB_STATISTICS view, 3-183
 USER_TAB_SUBPARTITIONS view, 3-184
 USER_TABLES view, 3-184
 TABS synonym, 3-154
 USER_TABLESPACES view, 3-184
 USER_TRANSFORMATIONS view, 3-184
 USER_TRIGGER_COLS view, 3-184
 USER_TRIGGERS view, 3-184
 USER_TS_QUOTAS view, 3-184
 USER_TUNE_MVIEW view, 3-184
 USER_TYPE_ATTRS view, 3-185
 USER_TYPE_METHODS view, 3-185
 USER_TYPE_VERSIONS view, 3-185
 USER_TYPES view, 3-185
 USER_UNUSED_COL_TABS view, 3-185
 USER_UPDATABLE_COLUMNS view, 3-185
 USER_USERS view, 3-185
 USER_USTATS view, 3-186
 USER_VARRAYS view, 3-186
 USER_VIEWS view, 3-186
 USER_WARNING_SETTINGS view, 3-186
 USER_WM_LOCKED_TABLES view, 2-3
 USER_WM_MODIFIED_TABLES view, 2-3
 USER_WM_PRIVS view, 2-3
 USER_WM_RIC_INFO view, 2-3
 USER_WM_TAB_TRIGGERS view, 2-3
 USER_WM_VERSIONED_TABLES view, 2-3
 USER_WM_VT_ERRORS view, 2-3
 USER_WORKSPACE_PRIVS view, 2-3
 USER_WORKSPACE_SAVEPOINTS view, 2-3
 USER_WORKSPACES view, 2-3
 USER_XML_SCHEMAS view, 3-186
 USER_XML_TAB_COLS view, 3-186
 USER_XML_TABLES view, 3-186
 USER_XML_VIEW_COLS view, 3-186
 USER_XML_VIEWS view, 3-187
 USERLOCK.SQL script, B-2
 users

authenticating, 1-100
 multiple, 1-50
 UTL_FILE_DIR initialization parameter, 1-144
 UTLBSTAT.SQL script, B-3
 UTLCHN1.SQL script, B-3
 UTLCONST.SQL script, B-3
 UTLDTREE.SQL script, 3-139, 3-145, B-3
 UTLESTAT.SQL script, B-3
 UTLEXPT1.SQL script, B-3
 UTLIP.SQL script, B-3
 UTLIRP.SQL script, B-3
 UTLLOCKT.SQL script, B-3
 UTLPWDMG.SQL script, B-3
 UTLRP.SQL script, B-3
 UTLSAMPL.SQL script, B-3
 UTLSCLN.SQL script, B-3
 UTLTKPRF.SQL script, B-4
 UTLU101I.SQL script, B-5
 UTLU101S.SQL script, B-5
 UTLVALID.SQL script, B-4
 UTLXPLAN.SQL script, B-4

V

V\$ACCESS view, 4-2
 V\$ACTIVE_INSTANCES view, 4-2
 V\$ACTIVE_SERVICES view, 4-3
 V\$ACTIVE_SESS_POOL_MTH view, 4-3
 V\$ACTIVE_SESSION_HISTORY view, 4-3
 V\$ALERT_TYPES view, 4-5
 V\$AQ view, 4-5
 V\$ARCHIVE view, 4-5
 V\$ARCHIVE_DEST view, 4-6
 V\$ARCHIVE_DEST_STATUS view, 4-8
 V\$ARCHIVE_GAP view, 4-9
 V\$ARCHIVE_PROCESSES view, 4-10
 V\$ARCHIVED_LOG view, 4-10
 V\$ASM_ALIAS view, 4-12
 V\$ASM_CLIENT view, 4-12
 V\$ASM_DISK view, 4-13
 V\$ASM_DISKGROUP view, 4-15
 V\$ASM_FILE view, 4-15
 V\$ASM_OPERATION view, 4-16
 V\$ASM_TEMPLATE view, 4-17
 V\$BACKUP view, 4-17
 V\$BACKUP_ASYNC_IO view, 4-17
 V\$BACKUP_CORRUPTION view, 4-18
 V\$BACKUP_DATAFILE view, 4-19
 V\$BACKUP_DEVICE view, 4-20
 V\$BACKUP_FILES view, 4-20
 V\$BACKUP_PIECE view, 4-22
 V\$BACKUP_REDOLOG view, 4-23
 V\$BACKUP_SET view, 4-24
 V\$BACKUP_SPFIL view, 4-25
 V\$BACKUP_SYNC_IO view, 4-25
 V\$BGPROCESS view, 4-26
 V\$BH view, 4-26
 V\$BLOCK_CHANGE_TRACKING view, 4-27
 V\$BUFFER_POOL view, 4-28
 V\$BUFFER_POOL_STATISTICS view, 4-28

V\$BUFFERED_PUBLISHERS view, 4-29
V\$BUFFERED_QUEUES view, 4-29
V\$BUFFERED_SUBSCRIBERS view, 4-30
V\$CACHE view, 4-31
V\$CACHE_LOCK view, 4-32
V\$CACHE_TRANSFER view, 4-34
V\$CIRCUIT view, 4-34
V\$CLASS_CACHE_TRANSFER view, 4-35
V\$CLIENT_STATS view, 4-36
V\$CONTEXT view, 4-36
V\$CONTROLFILE_RECORD_SECTION view, 4-36
V\$COPY_CORRUPTION view, 4-37
V\$CR_BLOCK_SERVER view, 4-38
V\$CURRENT_BLOCK_SERVER view, 4-39
V\$DATABASE view, 4-39
V\$DATABASE_BLOCK_CORRUPTION view, 4-43
V\$DATABASE_INCARNATION view, 4-43
V\$DATAFILE view, 4-44
V\$DATAFILE_COPY view, 4-45
V\$DATAFILE_HEADER view, 4-46
V\$DATAGUARD_CONFIG view, 4-47
V\$DATAGUARD_STATUS view, 4-47
V\$DB_CACHE_ADVICE view, 4-48
V\$DB_OBJECT_CACHE view, 4-48
V\$DB_PIPES view, 4-49
V\$DBFILE view, 4-49
V\$DBLINK view, 4-49
V\$DELETED_OBJECT view, 4-50
V\$DISPATCHER view, 4-50
V\$DISPATCHER_CONFIG view, 4-51
V\$DISPATCHER_RATE view, 4-52
V\$ENABLEDPRIVS view, 4-55
V\$ENQUEUE_LOCK view, 4-55
V\$ENQUEUE_STAT view, 4-56
V\$EVENT_HISTOGRAM view, 4-56
V\$EVENT_NAME view, 4-57
V\$EVENTMETRIC view, 4-57
V\$EXECUTION view, 4-57
V\$FALSE_PING view, 4-58
V\$FAST_START_SERVERS view, 4-58
V\$FAST_START_TRANSACTIONS view, 4-59
V\$FILE_CACHE_TRANSFER view, 4-59
V\$FILE_HISTOGRAM view, 4-60
V\$FILEMETRIC view, 4-60
V\$FILEMETRIC_HISTORY view, 4-60
V\$FILESTAT view, 4-61
V\$FIXED_TABLE view, 4-61
V\$FIXED_VIEW_DEFINITION view, 4-62
V\$FLASHBACK_DATABASE_LOG view, 4-62
V\$FLASHBACK_DATABASE_STAT view, 4-62
V\$GC_ELEMENT view, 4-62
V\$GC_ELEMENTS_WITH_COLLISIONS view, 4-63
V\$GCSHVMMASTER_INFO view, 4-63
V\$GCSPFMASTER_INFO view, 4-63
V\$GES_BLOCKING_ENQUEUE view, 4-64
V\$GES_CONVERT_LOCAL view, 4-65
V\$GES_CONVERT_REMOTE view, 4-65
V\$GES_ENQUEUE view, 4-65
V\$GES_LATCH view, 4-66
V\$GES_RESOURCE view, 4-66
V\$GES_STATISTICS view, 4-67
V\$GLOBAL_BLOCKED_LOCKS view, 4-67
V\$GLOBAL_TRANSACTION view, 4-67
V\$HS_AGENT view, 4-68
V\$HS_PARAMETER view, 4-68
V\$HS_SESSION view, 4-68
V\$HVMMASTER_INFO view, 4-69
V\$INDEXED_FIXED_COLUMN view, 4-69
V\$INSTANCE view, 4-69
V\$INSTANCE_RECOVERY view, 4-70
V\$JAVA_LIBRARY_CACHE_MEMORY view, 4-71
V\$JAVA_POOL_ADVICE view, 4-72
V\$LATCH view, 4-72
V\$LATCH_CHILDREN view, 4-73
V\$LATCH_MISSES view, 4-74
V\$LATCH_PARENT view, 4-74
V\$LATCHHOLDER view, 4-74
V\$LATCHNAME view, 4-74
V\$LIBRARY_CACHE_MEMORY view, 4-75
V\$LIBRARYCACHE view, 4-75
V\$LICENSE view, 4-76
V\$LOADISTAT view, 4-76
V\$LOADPSTAT view, 4-76
V\$LOCK view, 4-76
V\$LOCK_ACTIVITY view, 4-78
V\$LOCKED_OBJECT view, 4-78
V\$LOG view, 4-78
V\$LOG_HISTORY view, 4-79
V\$LOGFILE view, 4-79
V\$LOGHIST view, 4-80
V\$LOGMNR_CONTENTS view, 4-80
V\$LOGMNR_DICTIONARY view, 4-82
V\$LOGMNR_LOGS view, 4-82
V\$LOGMNR_PARAMETERS view, 4-83
V\$LOGSTDBY view, 4-83
V\$LOGSTDBY_STATS view, 4-84
V\$MANAGED_STANDBY view, 4-85
V\$MAP_COMP_LIST view, 4-86
V\$MAP_ELEMENT view, 4-86
V\$MAP_EXT_ELEMENT view, 4-87
V\$MAP_FILE view, 4-87
V\$MAP_FILE_EXTENT view, 4-87
V\$MAP_FILE_IO_STACK view, 4-88
V\$MAP_LIBRARY view, 4-88
V\$MAP_SUBELEMENT view, 4-89
V\$METRICNAME view, 4-89
V\$MTTR_TARGET_ADVICE view, 4-89
V\$MVREFRESH view, 4-90
V\$MYSTAT view, 4-90
V\$NLS_PARAMETERS view, 5-1
V\$NLS_VALID_VALUES view, 5-1
V\$OBJECT_DEPENDENCY view, 5-1
V\$OBJECT_USAGE view, 5-2
V\$OBSOLETE_BACKUP_FILES view, 5-2
V\$OBSOLETE_PARAMETER view, 5-4
V\$OFFLINE_RANGE view, 5-4
V\$OPEN_CURSOR view, 5-4
V\$OPTION view, 5-5
V\$OSSTAT view, 5-5

V\$PARALLEL_DEGREE_LIMIT_MTH view, 5-6
V\$PARAMETER view, 5-6
V\$PARAMETER2 view, 5-7
V\$PGA_TARGET_ADVICE view, 5-9
V\$PGA_TARGET_ADVICE_HISTOGRAM view, 5-9
V\$PGASTAT view, 5-10
V\$PQ_SESSTAT view, 5-11
V\$PQ_SLAVE view, 5-12
V\$PQ_SYSSTAT view, 5-12
V\$PQ_TQSTAT view, 5-13
V\$PROCESS view, 5-14
V\$PROPAGATION_RECEIVER view, 5-14
V\$PROPAGATION_SENDER view, 5-14
V\$PROXY_ARCHIVEDLOG view, 5-15
V\$PROXY_DATAFILE view, 5-16
V\$PWFILERS view, 5-17
V\$PX_PROCESS view, 5-17
V\$PX_PROCESS_SYSTAT view, 5-18
V\$PX_SESSION view, 5-18
V\$PX_SESSTAT view, 5-19
V\$QUEUE view, 5-19
V\$QUEUEING_MTH view, 5-20
V\$RECOVER_FILE view, 5-20
V\$RECOVERY_FILE_DEST view, 5-20
V\$RECOVERY_FILE_STATUS view, 5-20
V\$RECOVERY_LOG view, 5-21
V\$RECOVERY_PROGRESS view, 5-21
V\$RECOVERY_STATUS view, 5-22
V\$REPLPROP view, 5-22
V\$REPLQUEUE view, 5-23
V\$REQDIST view, 5-23
V\$RESERVED_WORDS view, 5-24
V\$RESOURCE view, 5-24
V\$RESOURCE_LIMIT view, 5-24
V\$RMAN_CONFIGURATION view, 5-25
V\$RMAN_OUTPUT view, 5-25
V\$RMAN_STATUS view, 5-26
V\$ROLLNAME view, 5-26
V\$ROLLSTAT view, 5-26
V\$ROWCACHE view, 5-27
V\$ROWCACHE_PARENT view, 5-28
V\$ROWCACHE_SUBORDINATE view, 5-28
V\$RSRC_CONSUMER_GROUP view, 5-29
V\$RSRC_CONSUMER_GROUP_CPU_MTH view, 5-29
V\$RSRC_PLAN view, 5-29
V\$RSRC_PLAN_CPU_MTH view, 5-30
V\$RULE view, 5-30
V\$RULE_SET view, 5-31
V\$RULE_SET_AGGREGATE_STATS view, 5-31
V\$SEGMENT_STATISTICS view, 5-32
V\$SEGSTAT view, 5-33
V\$SEGSTAT_NAME view, 5-33
V\$SERV_MOD_ACT_STATS view, 5-33
V\$SERVICE_EVENT view, 5-33
V\$SERVICE_STATS view, 5-34
V\$SERVICE_WAIT_CLASS view, 5-34
V\$SERVICEMETRIC view, 5-35
V\$SERVICEMETRIC_HISTORY view, 5-35
V\$SERVICES view, 5-35
V\$SES_OPTIMIZER_ENV view, 5-35
V\$SESS_IO view, 5-36
V\$SESS_TIME_MODEL view, 5-36
V\$SESSION view, 5-37
V\$SESSION_CONNECT_INFO view, 5-42
V\$SESSION_CURSOR_CACHE view, 5-42
V\$SESSION_EVENT view, 5-43, C-1
V\$SESSION_LONGOPS view, 5-43
V\$SESSION_OBJECT_CACHE view, 5-44
V\$SESSION_WAIT view, 5-45, C-1
V\$SESSION_WAIT_CLASS view, 5-46
V\$SESSION_WAIT_HISTORY view, 5-46
V\$SESSMETRIC view, 5-47
V\$SESSTAT view, 5-47, E-1
V\$SGA view, 5-47
V\$SGA_CURRENT_RESIZE_OPS view, 5-48
V\$SGA_DYNAMIC_COMPONENTS view, 5-48
V\$SGA_DYNAMIC_FREE_MEMORY view, 5-49
V\$SGA_RESIZE_OPS view, 5-49
V\$SGAINFO view, 5-50
V\$SGASTAT view, 5-50
V\$SHARED_POOL_ADVICE view, 5-50
V\$SHARED_POOL_RESERVED view, 5-51
V\$SHARED_SERVER view, 5-51
V\$SHARED_SERVER_MONITOR view, 5-52
V\$SORT_SEGMENT view, 5-52
V\$SPPARAMETER view, 5-53
V\$SQL view, 5-53
V\$SQL_BIND_CAPTURE view, 5-55
V\$SQL_BIND_DATA view, 5-56
V\$SQL_BIND_METADATA view, 5-57
V\$SQL_CURSOR view, 5-57
V\$SQL_OPTIMIZER_ENV view, 5-58
V\$SQL_PLAN view, 5-58
V\$SQL_PLAN_STATISTICS view, 5-60
V\$SQL_PLAN_STATISTICS_ALL view, 5-61
V\$SQL_REDIRECTION view, 5-63
V\$SQL_SHARED_CURSOR view, 5-64
V\$SQL_SHARED_MEMORY view, 5-65
V\$SQL_WORKAREA view, 5-65
V\$SQL_WORKAREA_ACTIVE view, 5-66
V\$SQL_WORKAREA_HISTOGRAM view, 5-67
V\$SQLAREA view, 5-68
V\$SQLTEXT view, 5-69
V\$SQLTEXT_WITH_NEWLINES view, 5-70
V\$STANDBY_LOG view, 5-70
V\$STATISTICS_LEVEL view, 5-71
V\$STATNAME view, 5-71, E-1
V\$STREAMS_APPLY_COORDINATOR view, 5-72
V\$STREAMS_APPLY_READER view, 5-73
V\$STREAMS_APPLY_SERVER view, 5-74
V\$STREAMS_CAPTURE view, 5-76
V\$SUBCACHE view, 5-77
V\$SYS_OPTIMIZER_ENV view, 5-78
V\$SYS_TIME_MODEL view, 5-78
V\$SYSAUX_OCCUPANTS view, 5-78
V\$SYSMETRIC view, 5-79
V\$SYSMETRIC_HISTORY view, 5-79
V\$SYSMETRIC_SUMMARY view, 5-79

- V\$SYSSTAT view, 5-80, E-1
- V\$SYSTEM_CURSOR_CACHE view, 5-80
- V\$SYSTEM_EVENT view, 5-80, C-1
- V\$SYSTEM_PARAMETER view, 5-81
- V\$SYSTEM_PARAMETER2 view, 5-82
- V\$SYSTEM_WAIT_CLASS view, 5-83
- V\$TABLESPACE view, 5-83
- V\$TEMP_CACHE_TRANSFER view, 5-84
- V\$TEMP_EXTENT_MAP view, 5-84
- V\$TEMP_EXTENT_POOL view, 5-84
- V\$TEMP_HISTOGRAM view, 5-85
- V\$TEMP_SPACE_HEADER view, 5-85
- V\$TEMPFILE view, 5-85
- V\$TEMPORARY_LOBS view, 5-86
- V\$TEMPSEG_USAGE view, 5-86
- V\$TEMPSTAT view, 5-87
- V\$THREAD view, 5-87
- V\$THRESHOLD_TYPES view, 5-88
- V\$TIMER view, 5-88
- V\$TIMEZONE_NAMES view, 5-88
- V\$TRANSACTION view, 5-88
- V\$TRANSACTION_ENQUEUE view, 5-89
- V\$TRANSPORTABLE_PLATFORM view, 5-90
- V\$TYPE_SIZE view, 5-90
- V\$UNDOSTAT view, 5-91
- V\$VERSION view, 5-92
- V\$VPD_POLICY view, 5-92
- V\$WAITCLASSMETRIC view, 5-92
- V\$WAITCLASSMETRIC_HISTORY view, 5-93
- V\$WAITSTAT view, 5-93
- views
 - data dictionary views, 2-1
 - data files, 1-56

W

wait events

- alter system set dispatchers, C-13
- batched allocate scn lock request, C-13
- BFILE check if exists, C-13
- BFILE check if open, C-13
- BFILE closure, C-13
- BFILE get length, C-13
- BFILE get name object, C-14
- BFILE get path object, C-14
- BFILE internal seek, C-14
- BFILE open, C-14
- BFILE read, C-14
- buffer busy waits, C-15
- buffer deadlock, C-15
- buffer for checkpoint, C-15
- buffer latch, C-16
- buffer read retry, C-16
- checkpoint completed, C-16
- checkpoint range buffer not saved, C-16
- controlfile parallel write, C-17
- controlfile sequential read, C-17
- controlfile single write, C-17
- conversion file read, C-18
- db file parallel read, C-18

- db file parallel write, C-18
- db file scattered read, C-18
- db file sequential read, C-19
- db file single write, C-19
- DFS db file lock, C-19
- DFS lock handle, C-20
- direct path read, C-20
- direct path write, C-20
- dispatcher shutdown, C-21
- dispatcher timer, C-21
- duplicate cluster key, C-21
- enqueue, C-21
- file identify, C-22
- file open, C-22
- free buffer waits, C-22
- free global transaction table entry, C-22
- free process state object, C-22
- global cache busy, C-22
- global cache freelist wait, C-23
- global cache lock cleanup, C-23
- global cache null to s, C-23
- global cache null to x, C-23
- global cache open s, C-24
- global cache open x, C-24
- global cache s to x, C-24
- inactive session, C-24
- inactive transaction branch, C-25
- index block split, C-25
- instance recovery, C-25
- instance state change, C-25
- io done, C-26
- kcl bg acks, C-26
- latch activity, C-26
- latch free, C-26
- library cache load lock, C-27
- library cache lock, C-27
- library cache pin, C-28
- lock manager wait for remote message, C-28
- log buffer space, C-28
- log file parallel write, C-28
- log file sequential read, C-28
- log file single write, C-29
- log file switch (archiving needed), C-29
- log file switch (checkpoint incomplete), C-29
- log file switch (clearing log file), C-29
- log file switch completion, C-29
- log file sync, C-30
- log switch/archive, C-30
- on-going SCN fetch to complete, C-30
- pending global transaction(s), C-30
- pipe get, C-30
- pipe put, C-31
- PL/SQL lock timer, C-31
- pmon timer, C-31
- process startup, C-31
- PX dequeue wait, C-31
- PX qref latch, C-32
- PX server shutdown, C-32
- PX signal server, C-32
- queue messages, C-32

- rdbms ipc message, C-33
- rdbms ipc message block, C-33
- rdbms ipc reply, C-33
- redo wait, C-33
- row cache lock, C-33
- scginq AST call, C-34
- single-task message, C-34
- smon timer, C-34
- SQL*Net break/reset to client, C-34
- SQL*Net break/reset to dblink, C-34
- SQL*Net message from client, C-35
- SQL*Net message from dblink, C-35
- SQL*Net message to client, C-35
- SQL*Net message to dblink, C-35
- SQL*Net more data from client, C-35
- SQL*Net more data from dblink, C-36
- SQL*Net more data to client, C-36
- SQL*Net more data to dblink, C-36
- switch logfile command, C-36
- timer in sksawat, C-36
- transaction, C-37
- unbound tx, C-37
- undo segment extension, C-37
- undo segment recovery, C-37
- undo segment tx slot, C-37
- virtual circuit status, C-38
- WMON goes to sleep, C-38
- write complete waits, C-38
- writes stopped by instance recovery or database suspension, C-38
- WM_INSTALLATION view, 2-3
- WM_REPLICATION_INFO view, 2-3
- WORKAREA_SIZE_POLICY initialization parameter, 1-145

