Oracle9*i*

Database Reference

Release 2 (9.2)

March 2002

Part No. A96536-01

ORACLE®

Oracle9*i* Database Reference, Release 2 (9.2)

Part No. A96536-01

Primary Author: Tony Morales

Contributing Author: Diana Lorentz

Contributors: Nipun Agarwal, Angela Amor, Rick Anderson, Lance Ashdown, Mark Bauer, Ruth Baylis, Neerja Bhatt, Tudor Bosman, George Candea, Wilson Chan, Sumanta Chatterjee, Eugene Chong, Michele Cyran, Mary Ann Davidson, 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, Nina Lewis, Yunrui Li, Juan Loaiza, Neil MacNaughton, Kant Patel, Shankar Raman, Ari Mozes, Subramanian Muralidhar, Ravi Murthy, Karuna Muthiah, Arvind Nithrakashyap, Ronald Obermarck, Cetin Ozbutun, Christopher Racicot, Mark Ramacher, Suvendu Ray, Ann Rhee, Vivian Schupmann, Harlan Sexton, Helen Slattery, Ekrem Soylemez, 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

Contents

2 Static Data Dictionary Views

3 Dynamic Performance (V$) Views

4 Database Limits

5 SQL Scripts

A Oracle Wait Events

Index

Send Us Your Comments

Oracle9*i* Database Reference, Release 2 (9.2)

Part No. A96536-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. 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?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, 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 (optionally) electronic mail address.

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 system.

Oracle9*i* Database Reference contains information that describes the features and functionality of the Oracle9*i* (also known as the standard edition) and the Oracle9*i* Enterprise Edition products. Oracle9*i* and the Oracle9*i* 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 Oracle9*i* Real Application Clusters option.

> **See Also:** *Oracle9i Database New Features* for information about the differences between Oracle9*i* and the Oracle9*i* Enterprise Edition and the features and options that are available to you.

This preface contains these topics:

Audience

Oracle9*i* 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

Organization

This document contains:

Chapter 1, "Initialization Parameters"

This chapter describes the database initialization parameters you can specify in the parameter file to start or configure an instance.

Chapter 2, "Static Data Dictionary Views"

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

Chapter 3, "Dynamic Performance (V$) Views"

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

Chapter 4, "Database Limits"

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

Chapter 5, "SQL Scripts"

This chapter describes the SQL scripts that are required for optimal operation of the Oracle database server.

Appendix A, "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 B, "Oracle Enqueue Names"

This appendix lists some enqueues used by Oracle9*i*.

Appendix C, "Statistics Descriptions"

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

Related Documentation

For more information, see these Oracle resources:

- *Oracle9i Database Concepts* for a comprehensive introduction to the concepts and terminology used in this manual
- *Oracle9i Database Administrator's Guide* for information about administering the Oracle database server
- *Oracle9i Database Migration* for the procedures for upgrading a previous release of Oracle to the new Oracle9*i* release
- *Oracle9i SQL Reference* for information on Oracle's SQL commands and functions
- *Oracle9i Application Developer's Guide - Fundamentals* for information about developing database applications within the Oracle database server

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 *Oracle9i Sample Schemas* for information on how these schemas were created and how you can use them yourself.

In North America, printed documentation is available for sale in the Oracle Store at

```
http://oraclestore.oracle.com/
```

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

```
http://www.oraclebookshop.com/
```

Other customers can contact their Oracle representative to purchase printed documentation.

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/admin/account/membership.html
```

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/docs/index.htm`

To access the database documentation search engine directly, please visit

`http://tahiti.oracle.com`

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 Microsoft 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**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* 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.

Convention	Meaning	Example
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 open 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.
lowercase monospace (fixed-width font) italic	Lowercase monospace italic font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* 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 (`*`digits`*` [ , `*`precision`*` ])`
{ }	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]`

Convention	Meaning	Example
. . .	Horizontal ellipsis points indicate either: ■ 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 *subquery*; SELECT *col1*, *col2*, ... , *coln* 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 /fs1/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;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
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 Microsoft Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > 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 *command prompt* in this manual.	`C:\oracle\oradata>`
	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)`
HOME_NAME	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`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default used one of the following names: ■ `C:\orant` for Windows NT ■ `C:\orawin98` for Windows 98 This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install Oracle9*i* release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora90`. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle9i Database Getting Started for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

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 Corporation 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

What's New in Oracle9*i* Database Reference?

This section describes new features of Oracle9*i* release 2 (9.2) and provides pointers to additional information. New features information from previous releases is also retained to help those users upgrading to the current release.

The following sections describe the new features in Oracle9*i* Database Reference:

Oracle9*i* Release 2 (9.2) New Features in Oracle9*i* Database Reference

The following initialization parameters are new to this release:

AUDIT_SYS_OPERATIONS
DG_BROKER_CONFIG_FILE*n*
DG_BROKER_START
FILE_MAPPING
FILESYSTEMIO_OPTIONS
LOG_PARALLELISM
OLAP_PAGE_POOL_SIZE
OPTIMIZER_DYNAMIC_SAMPLING
STATISTICS_LEVEL

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
ALL_APPLY	DBA_APPLY	
ALL_APPLY_CONFLICT_COLUMNS	DBA_APPLY_CONFLICT_COLUMNS	
ALL_APPLY_DML_HANDLERS	DBA_APPLY_DML_HANDLERS	
ALL_APPLY_ERROR	DBA_APPLY_ERROR	
	DBA_APPLY_INSTANTIATED_ OBJECTS	
ALL_APPLY_KEY_COLUMNS	DBA_APPLY_KEY_COLUMNS	
ALL_APPLY_PARAMETERS	DBA_APPLY_PARAMETERS	
ALL_APPLY_PROGRESS	DBA_APPLY_PROGRESS	
	DBA_AQ_AGENT_PRIVS	USER_AQ_AGENT_PRIVS
	DBA_AQ_AGENTS	
	DBA_AW_PS	USER_AW_PS
	DBA_AWS	USER_AWS
ALL_CAPTURE	DBA_CAPTURE	
ALL_CAPTURE_PARAMETERS	DBA_CAPTURE_PARAMETERS	
ALL_CAPTURE_PREPARED_DATABASE	DBA_CAPTURE_PREPARED_DATABASE	
ALL_CAPTURE_PREPARED_SCHEMAS	DBA_CAPTURE_PREPARED_SCHEMAS	
ALL_CAPTURE_PREPARED_TABLES	DBA_CAPTURE_PREPARED_TABLES	
ALL_CONS_OBJ_COLUMNS	DBA_CONS_OBJ_COLUMNS	USER_CONS_OBJ_COLUMNS

ALL_ Views	DBA_ Views	USER_ Views
ALL_EVALUATION_CONTEXT_TABLES	DBA_EVALUATION_CONTEXT_TABLES	USER_EVALUATION_CONTEXT_ TABLES
ALL_EVALUATION_CONTEXT_VARS	DBA_EVALUATION_CONTEXT_VARS	USER_EVALUATION_CONTEXT_VARS
ALL_EVALUATION_CONTEXTS	DBA_EVALUATION_CONTEXTS	USER_EVALUATION_CONTEXTS
ALL_JAVA_ARGUMENTS	DBA_JAVA_ARGUMENTS	USER_JAVA_ARGUMENTS
ALL_JAVA_CLASSES	DBA_JAVA_CLASSES	USER_JAVA_CLASSES
ALL_JAVA_DERIVATIONS	DBA_JAVA_DERIVATIONS	USER_JAVA_DERIVATIONS
ALL_JAVA_FIELDS	DBA_JAVA_FIELDS	USER_JAVA_FIELDS
ALL_JAVA_IMPLEMENTS	DBA_JAVA_IMPLEMENTS	USER_JAVA_IMPLEMENTS
ALL_JAVA_INNERS	DBA_JAVA_INNERS	USER_JAVA_INNERS
ALL_JAVA_LAYOUTS	DBA_JAVA_LAYOUTS	USER_JAVA_LAYOUTS
ALL_JAVA_METHODS	DBA_JAVA_METHODS	USER_JAVA_METHODS
ALL_JAVA_NCOMPS	DBA_JAVA_NCOMPS	USER_JAVA_NCOMPS
ALL_JAVA_RESOLVERS	DBA_JAVA_RESOLVERS	USER_JAVA_RESOLVERS
ALL_JAVA_THROWS	DBA_JAVA_THROWS	USER_JAVA_THROWS
ALL_LOB_TEMPLATES	DBA_LOB_TEMPLATES	USER_LOB_TEMPLATES
	DBA_LOGSTDBY_EVENTS	
	DBA_LOGSTDBY_LOG	
	DBA_LOGSTDBY_NOT_UNIQUE	
	DBA_LOGSTDBY_PARAMETERS	
	DBA_LOGSTDBY_PROGRESS	
	DBA_LOGSTDBY_SKIP	
	DBA_LOGSTDBY_SKIP_TRANSACTION	
	DBA_LOGSTDBY_UNSUPPORTED	
ALL_OBJ_COLATTRS	DBA_OBJ_COLATTRS	USER_OBJ_COLATTRS
ALL_PROPAGATION	DBA_PROPAGATION	
ALL_PUBLISHED_COLUMNS	DBA_PUBLISHED_COLUMNS	USER_PUBLISHED_COLUMNS
	DBA_REGISTRY	USER_REGISTRY
ALL_REGISTRY_BANNERS		
	DBA_REGISTRY_HIERARCHY	
ALL_RULE_SET_RULES	DBA_RULE_SET_RULES	USER_RULE_SET_RULES
ALL_RULE_SETS	DBA_RULE_SETS	USER_RULE_SETS

ALL_ Views	DBA_ Views	USER_ Views
ALL_RULES	DBA_RULES	USER_RULES
ALL_STREAMS_GLOBAL_RULES	DBA_STREAMS_GLOBAL_RULES	
ALL_STREAMS_SCHEMA_RULES	DBA_STREAMS_SCHEMA_RULES	
ALL_STREAMS_TABLE_RULES	DBA_STREAMS_TABLE_RULES	
ALL_SUBPARTITION_TEMPLATES	DBA_SUBPARTITION_TEMPLATES	USER_SUBPARTITION_TEMPLATES

See Also: Chapter 2, "Static Data Dictionary Views" for more information about these static data dictionary views

The following dynamic performance views are new to this release:

GV$AW_CALC	V$AW_CALC
GV$AW_OLAP	V$AW_OLAP
GV$AW_SESSION_INFO	V$AW_SESSION_INFO
GV$BACKUP_SPFILE	V$BACKUP_SPFILE
GV$DATABASE_BLOCK_CORRUPTION	V$DATABASE_BLOCK_CORRUPTION
GV$DATABASE_INCARNATION	V$DATABASE_INCARNATION
GV$DATAGUARD_STATUS	V$DATAGUARD_STATUS
GV$LIBRARY_CACHE_MEMORY	V$LIBRARY_CACHE_MEMORY
GV$LOGMNR_STATS	V$LOGMNR_STATS
GV$MAP_COMP_LIST	V$MAP_COMP_LIST
GV$MAP_ELEMENT	V$MAP_ELEMENT
GV$MAP_EXT_ELEMENT	V$MAP_EXT_ELEMENT
GV$MAP_FILE	V$MAP_FILE
GV$MAP_FILE_EXTENT	V$MAP_FILE_EXTENT
GV$MAP_FILE_IO_STACK	V$MAP_FILE_IO_STACK
GV$MAP_LIBRARY	V$MAP_LIBRARY
GV$MAP_SUBELEMENT	V$MAP_SUBELEMENT
GV$MTTR_TARGET_ADVICE	V$MTTR_TARGET_ADVICE
GV$PGA_TARGET_ADVICE	V$PGA_TARGET_ADVICE

GV$PGA_TARGET_ADVICE_HISTOGRAM	V$PGA_TARGET_ADVICE_HISTOGRAM
GV$SEGMENT_STATISTICS	V$SEGMENT_STATISTICS
GV$SEGSTAT	V$SEGSTAT
GV$SEGSTAT_NAME	V$SEGSTAT_NAME
GV$SGA_CURRENT_RESIZE_OPS	V$SGA_CURRENT_RESIZE_OPS
GV$SGA_DYNAMIC_COMPONENTS	V$SGA_DYNAMIC_COMPONENTS
GV$SGA_DYNAMIC_FREE_MEMORY	V$SGA_DYNAMIC_FREE_MEMORY
GV$SGA_RESIZE_OPS	V$SGA_RESIZE_OPS
GV$SHARED_POOL_ADVICE	V$SHARED_POOL_ADVICE
GV$SQL_PLAN_STATISTICS	V$SQL_PLAN_STATISTICS
GV$SQL_PLAN_STATISTICS_ALL	V$SQL_PLAN_STATISTICS_ALL
GV$SQL_WORKAREA_HISTOGRAM	V$SQL_WORKAREA_HISTOGRAM
GV$STATISTICS_LEVEL	V$STATISTICS_LEVEL
GV$STREAMS_APPLY_COORDINATOR	V$STREAMS_APPLY_COORDINATOR
GV$STREAMS_APPLY_READER	V$STREAMS_APPLY_READER
GV$STREAMS_APPLY_SERVER	V$STREAMS_APPLY_SERVER
GV$STREAMS_CAPTURE	V$STREAMS_CAPTURE
GV$TEMPSEG_USAGE	V$TEMPSEG_USAGE

See Also: Chapter 3, "Dynamic Performance (V$) Views" 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_IND_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
ALL_LOBS	RETENTION
	FREEPOOLS

Static Data Dictionary View	New Columns
ALL_MVIEWS	UNKNOWN_PREBUILT
	UNKNOWN_PLSQL_FUNC
	UNKNOWN_EXTERNAL_TABLE
	UNKNOWN_CONSIDER_FRESH
	UNKNOWN_IMPORT
ALL_NESTED_TABLES	ELEMENT_SUBSTITUTABLE
ALL_PART_TABLES	DEF_COMPRESSION
ALL_QUEUE_TABLES	SECURE
ALL_TAB_PARTITIONS	COMPRESSION
ALL_TAB_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
	COMPRESSION
ALL_VARRAYS	ELEMENT_SUBSTITUTABLE
DBA_FGA_AUDIT_TRAIL	USERHOST
	CLIENT_ID
	EXT_NAME
DBA_IND_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
DBA_LOBS	RETENTION
	FREEPOOLS
DBA_MVIEWS	UNKNOWN_PREBUILT
	UNKNOWN_PLSQL_FUNC
	UNKNOWN_EXTERNAL_TABLE
	UNKNOWN_CONSIDER_FRESH
	UNKNOWN_IMPORT
DBA_NESTED_TABLES	ELEMENT_SUBSTITUTABLE
DBA_PART_TABLES	DEF_COMPRESSION
DBA_QUEUE_TABLES	SECURE
DBA_TAB_PARTITIONS	COMPRESSION

Static Data Dictionary View	New Columns
DBA_TAB_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
	COMPRESSION
DBA_TABLESPACES	FORCE_LOGGING
DBA_VARRAYS	ELEMENT_SUBSTITUTABLE
USER_IND_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
USER_LOBS	RETENTION
	FREEPOOLS
USER_MVIEWS	UNKNOWN_PREBUILT
	UNKNOWN_PLSQL_FUNC
	UNKNOWN_EXTERNAL_TABLE
	UNKNOWN_CONSIDER_FRESH
	UNKNOWN_IMPORT
USER_NESTED_TABLES	ELEMENT_SUBSTITUTABLE
USER_PART_TABLES	DEF_COMPRESSION
USER_QUEUE_TABLES	SECURE
USER_TAB_PARTITIONS	COMPRESSION
USER_TAB_SUBPARTITIONS	HIGH_VALUE
	HIGH_VALUE_LENGTH
	COMPRESSION
USER_TABLESPACES	FORCE_LOGGING
USER_VARRAYS	ELEMENT_SUBSTITUTABLE

See Also: Chapter 2, "Static Data Dictionary 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	DEST_NAME
	NET_TIMEOUT
	REGISTER
	REMOTE_TEMPLATE
GV$ARCHIVE_DEST_STATUS and V$ARCHIVE_DEST_STATUS	DEST_NAME
	PROTECTION_MODE
	SRL
GV$ARCHIVED_LOG and V$ARCHIVED_LOG	ACTIVATION#
GV$BACKUP_CORRUPTION and V$BACKUP_CORRUPTION	CORRUPTION_TYPE
GV$COPY_CORRUPTION and V$COPY_CORRUPTION	CORRUPTION_TYPE
GV$DATABASE and V$DATABASE	PROTECTION_MODE
	PROTECTION_LEVEL
	DATAGUARD_BROKER
	GUARD_STATUS
	SUPPLEMENTAL_LOG_DATA_MIN
	SUPPLEMENTAL_LOG_DATA_PK
	SUPPLEMENTAL_LOG_DATA_UI
	FORCE_LOGGING
GV$DATAFILE_COPY and V$DATAFILE_COPY	SCANNED
GV$DB_CACHE_ADVICE and V$DB_CACHE_ADVICE	SIZE_FACTOR
GV$DB_OBJECT_CACHE and V$DB_OBJECT_CACHE	CHILD_LATCH
GV$FILE_CACHE_TRANSFER and V$FILE_CACHE_TRANSFER	CR_TRANSFERS
	CUR_TRANSFERS
GV$LATCH_CHILDREN and V$LATCH_CHILDREN	MISSES_WAITLIST
	SLEEPS_WAITLIST
	WAIT_TIME_WAITLIST

Dynamic Performance View	New Columns
GV$LATCH_PARENT and V$LATCH_PARENT	MISSES_WAITLIST
	SLEEPS_WAITLIST
	WAIT_TIME_WAITLIST
GV$LOGMNR_CALLBACK and V$LOGMNR_CALLBACK	STATE
	TYPE
	CAPABILITY
GV$LOGMNR_CONTENTS and V$LOGMNR_CONTENTS	PXIDUSN
	PXIDSLT
	PXIDSQN
GV$LOGMNR_REGION and V$LOGMNR_REGION	MEMSTATE
	STATE
GV$LOGSTDBY and V$LOGSTDBY	HIGH_SCN
GV$PGASTAT and V$PGASTAT	UNIT
GV$PROCESS and V$PROCESS	PGA_FREEABLE_MEM
GV$ROLLSTAT and V$ROLLSTAT	LATCH
GV$SQL and V$SQL	FETCHES
	CHILD_LATCH
GV$SQL_PLAN and V$SQL_PLAN	SEARCH_COLUMNS
	ACCESS_PREDICATES
	FILTER_PREDICATES
GV$SQL_SHARED_CURSOR and V$SQL_SHARED_CURSOR	NO_TRIGGER_MISMATCH
	FLASHBACK_CURSOR
GV$SQL_WORKAREA and V$SQL_WORKAREA	MAX_TEMPSEG_SIZE
	LAST_TEMPSEG_SIZE
GV$SQL_WORKAREA_ACTIVE and V$SQL_WORKAREA_ACTIVE	TEMPSEG_SIZE
	TABLESPACE
	SEGRFNO#
	SEGBLK#
GV$SQLAREA and V$SQLAREA	FETCHES
	CHILD_LATCH

See Also: Chapter 3, "Dynamic Performance (V$) Views" for more information about the new columns in these dynamic performance views

Oracle9*i* Release 1 (9.0.1) New Features in Oracle9*i* Database Reference

The following initialization parameters are new to this release:

ARCHIVE_LAG_TARGET	CIRCUITS
CLUSTER_INTERCONNECTS	CLUSTER_DATABASE
CLUSTER_DATABASE_INSTANCES	DB_*n*K_CACHE_SIZE
DB_CACHE_ADVICE	DB_CACHE_SIZE
DB_CREATE_FILE_DEST	DB_CREATE_ONLINE_LOG_DEST_*n*
DB_KEEP_CACHE_SIZE	DB_RECYCLE_CACHE_SIZE
DISPATCHERS	DRS_START
FAL_CLIENT	FAL_SERVER
FAST_START_MTTR_TARGET	GLOBAL_CONTEXT_POOL_SIZE
LOG_ARCHIVE_DEST_*n* (*n* = 6, 7, ... 10)	LOG_ARCHIVE_DEST_STATE_*n* (*n* = 6, 7, ... 10)
LOGMNR_MAX_PERSISTENT_SESSIONS	MAX_DISPATCHERS
MAX_SHARED_SERVERS	NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP	NLS_TIMESTAMP_FORMAT
NLS_TIMESTAMP_TZ_FORMAT	PGA_AGGREGATE_TARGET
PLSQL_COMPILER_FLAGS	PLSQL_NATIVE_C_COMPILER
PLSQL_NATIVE_LIBRARY_DIR	PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
PLSQL_NATIVE_LINKER	PLSQL_NATIVE_MAKE_FILE_NAME
PLSQL_NATIVE_MAKE_UTILITY	REMOTE_ARCHIVE_ENABLE
REMOTE_LISTENER	SGA_MAX_SIZE
SHARED_SERVER_SESSIONS	SHARED_SERVERS
SPFILE	STANDBY_FILE_MANAGEMENT

STANDBY_PRESERVES_NAMES	trace_enabled
UNDO_MANAGEMENT	UNDO_RETENTION
UNDO_SUPPRESS_ERRORS	UNDO_TABLESPACE
WORKAREA_SIZE_POLICY	

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_APPLICATION_ROLES	USER_APPLICATION_ROLES
	DBA_ATTRIBUTE_TRANSFORMATIONS	USER_ATTRIBUTE_ TRANSFORMATIONS
ALL_AUDIT_POLICIES	DBA_AUDIT_POLICIES	USER_AUDIT_POLICIES
ALL_BASE_TABLE_MVIEWS	DBA_BASE_TABLE_MVIEWS	USER_BASE_TABLE_MVIEWS
ALL_EXTERNAL_LOCATIONS	DBA_EXTERNAL_LOCATIONS	USER_EXTERNAL_LOCATIONS
ALL_EXTERNAL_TABLES	DBA_EXTERNAL_TABLES	USER_EXTERNAL_TABLES
ALL_INDEXTYPE_COMMENTS	DBA_INDEXTYPE_COMMENTS	USER_INDEXTYPE_COMMENTS
ALL_JOIN_IND_COLUMNS	DBA_JOIN_IND_COLUMNS	USER_JOIN_IND_COLUMNS
	DBA_FGA_AUDIT_TRAIL	
	DBA_GLOBAL_CONTEXT	
ALL_LOG_GROUP_COLUMNS	DBA_LOG_GROUP_COLUMNS	USER_LOG_GROUP_COLUMNS
ALL_LOG_GROUPS	DBA_LOG_GROUPS	USER_LOG_GROUPS
	DBA_MVIEW_LOG_FILTER_COLS	
ALL_MVIEW_LOGS	DBA_MVIEW_LOGS	USER_MVIEW_LOGS
ALL_MVIEW_REFRESH_TIMES	DBA_MVIEW_REFRESH_TIMES	USER_MVIEW_REFRESH_TIMES
ALL_OPERATOR_COMMENTS	DBA_OPERATOR_COMMENTS	USER_OPERATOR_COMMENTS
ALL_PENDING_CONV_TABLES	DBA_PENDING_CONV_TABLES	USER_PENDING_CONV_TABLES
ALL_POLICY_CONTEXTS	DBA_POLICY_CONTEXTS	USER_POLICY_CONTEXTS
ALL_POLICY_GROUPS	DBA_POLICY_GROUPS	USER_POLICY_GROUPS
ALL_PROCEDURES	DBA_PROCEDURES	USER_PROCEDURES
	DBA_PROXIES	USER_PROXIES
	DBA_REGISTERED_MVIEW_GROUPS	

ALL_ Views	DBA_ Views	USER_ Views
ALL_REGISTERED_MVIEWS	DBA_REGISTERED_MVIEWS	USER_REGISTERED_MVIEWS
	DBA_REPCAT_EXCEPTIONS	
	DBA_REPEXTENSIONS	
	DBA_REPSITES_NEW	
	DBA_RESUMABLE	USER_RESUMABLE
ALL_SECONDARY_OBJECTS	DBA_SECONDARY_OBJECTS	USER_SECONDARY_OBJECTS
ALL_SOURCE_TAB_COLUMNS	DBA_SOURCE_TAB_COLUMNS	USER_SOURCE_TAB_COLUMNS
ALL_SOURCE_TABLES	DBA_SOURCE_TABLES	USER_SOURCE_TABLES
ALL_SQLJ_TYPE_ATTRS	DBA_SQLJ_TYPE_ATTRS	USER_SQLJ_TYPE_ATTRS
ALL_SQLJ_TYPE_METHODS	DBA_SQLJ_TYPE_METHODS	USER_SQLJ_TYPE_METHODS
ALL_SQLJ_TYPES	DBA_SQLJ_TYPES	USER_SQLJ_TYPES
ALL_STORED_SETTINGS	DBA_STORED_SETTINGS	USER_STORED_SETTINGS
ALL_SUBSCRIBED_COLUMNS	DBA_SUBSCRIBED_COLUMNS	USER_SUBSCRIBED_COLUMNS
ALL_SUBSCRIBED_TABLES	DBA_SUBSCRIBED_TABLES	USER_SUBSCRIBED_TABLES
ALL_SUBSCRIPTIONS	DBA_SUBSCRIPTIONS	USER_SUBSCRIPTIONS
ALL_TAB_COLS	DBA_TAB_COLS	USER_TAB_COLS
	DBA_TEMPLATE_REFGROUPS	
	DBA_TEMPLATE_TARGETS	
	DBA_TRANSFORMATIONS	USER_TRANSFORMATIONS
ALL_TYPE_VERSIONS	DBA_TYPE_VERSIONS	USER_TYPE_VERSIONS
	DBA_UNDO_EXTENTS	

See Also: Chapter 2, "Static Data Dictionary Views" for more information about these static data dictionary views

The following dynamic performance views are new to this release:

GV$ACTIVE_SESS_POOL_MTH	V$ACTIVE_SESS_POOL_MTH
GV$ARCHIVE_DEST_STATUS	V$ARCHIVE_DEST_STATUS
GV$ARCHIVE_GAP	V$ARCHIVE_GAP
GV$CACHE_TRANSFER	V$CACHE_TRANSFER
GV$CLASS_CACHE_TRANSFER	V$CLASS_CACHE_TRANSFER

GV$CR_BLOCK_SERVER	V$CR_BLOCK_SERVER
GV$ENQUEUE_STAT	V$ENQUEUE_STAT
GV$FILE_CACHE_TRANSFER	V$FILE_CACHE_TRANSFER
GV$GC_ELEMENT	V$GC_ELEMENT
GV$GC_ELEMENTS_WITH_COLLISIONS	V$GC_ELEMENTS_WITH_COLLISIONS
GV$GCSHVMASTER_INFO	V$GCSHVMASTER_INFO
GV$GCSPFMASTER_INFO	V$GCSPFMASTER_INFO
GV$GES_BLOCKING_ENQUEUE	V$GES_BLOCKING_ENQUEUE
GV$GES_CONVERT_LOCAL	V$GES_CONVERT_LOCAL
GV$GES_CONVERT_REMOTE	V$GES_CONVERT_REMOTE
GV$GES_ENQUEUE	V$GES_ENQUEUE
GV$GES_LATCH	V$GES_LATCH
GV$GES_RESOURCE	V$GES_RESOURCE
GV$GES_STATISTICS	V$GES_STATISTICS
GV$GES_TRAFFIC_CONTROLLER	V$GES_TRAFFIC_CONTROLLER
GV$GLOBALCONTEXT	V$GLOBALCONTEXT
GV$HVMASTER_INFO	V$HVMASTER_INFO
GV$LOGMNR_CALLBACK	V$LOGMNR_CALLBACK
GV$LOGMNR_LOGFILE	V$LOGMNR_LOGFILE
GV$LOGMNR_PROCESS	V$LOGMNR_PROCESS
GV$LOGMNR_REGION	V$LOGMNR_REGION
GV$LOGMNR_SESSION	V$LOGMNR_SESSION
GV$LOGMNR_TRANSACTION	V$LOGMNR_TRANSACTION
GV$LOGSTDBY	V$LOGSTDBY
GV$LOGSTDBY_STATS	V$LOGSTDBY_STATS
GV$MANAGED_STANDBY	V$MANAGED_STANDBY
GV$MVREFRESH	V$MVREFRESH
	V$OBJECT_USAGE
GV$PGASTAT	V$PGASTAT

GV$QUEUEING_MTH	V$QUEUEING_MTH
GV$REPLPROP	V$REPLPROP
GV$REPLQUEUE	V$REPLQUEUE
GV$RESUMABLE	V$RESUMABLE
GV$RMAN_CONFIGURATION	V$RMAN_CONFIGURATION
GV$SHARED_SERVER_MONITOR	V$SHARED_SERVER_MONITOR
GV$SPPARAMETER	V$SPPARAMETER
GV$SQL_PLAN	V$SQL_PLAN
GV$SQL_REDIRECTION	V$SQL_REDIRECTION
GV$SQL_WORKAREA	V$SQL_WORKAREA
GV$SQL_WORKAREA_ACTIVE	V$SQL_WORKAREA_ACTIVE
GV$STANDBY_LOG	V$STANDBY_LOG
GV$TEMP_CACHE_TRANSFER	V$TEMP_CACHE_TRANSFER
GV$TIMEZONE_NAMES	V$TIMEZONE_NAMES
GV$UNDOSTAT	V$UNDOSTAT
GV$VPD_POLICY	V$VPD_POLICY

See Also: Chapter 3, "Dynamic Performance (V$) Views" 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	DEPENDENCIES
ALL_ARGUMENTS	CHAR_LENGTH
	CHAR_USED
ALL_ASSOCIATIONS	INTERFACE_VERSION
ALL_CLUSTERS	DEPENDENCIES

Static Data Dictionary View	New Columns
ALL_CONSTRAINTS	INDEX_OWNER
	INDEX_NAME
	INVALID
	VIEW_RELATED
ALL_DIMENSIONS	COMPILE_STATE
ALL_IND_COLUMNS	CHAR_LENGTH
ALL_IND_PARTITIONS	DOMIDX_OPSTATUS
	PARAMETERS
ALL_INDEXES	JOIN_INDEX
ALL_INDEXTYPE_OPERATORS	OPERATOR_SCHEMA
ALL_INDEXTYPES	INTERFACE_VERSION
	PARTITIONING
ALL_MVIEWS	USE_NO_INDEX
ALL_OBJECT_TABLES	DEPENDENCIES
ALL_OPBINDINGS	PROPERTY
ALL_OUTLINES	SIGNATURE
ALL_PART_INDEXES	DEF_PARAMETERS
ALL_POLICIES	POLICY_GROUP
	STATIC_POLICY
ALL_REPCAT_TEMPLATE_OBJECTS	DDL_NUM
ALL_REPCAT_TEMPLATE_SITES	INSTANTIATION_DATE
ALL_REPCOLUMN	CTYPE_TOID
	CTYPE_OWNER
	CTYPE_HASHCODE
	CTYPE_MOD
	TOP
	CHAR_LENGTH
	CHAR_USED
ALL_REPDDL	DDL_NUM

Static Data Dictionary View	New Columns
ALL_REPFLAVOR_COLUMNS	TYPE_TOID
	TYPE_OWNER
	TYPE_HASHCODE
	TYPE_MOD
	TOP
ALL_REPOBJECT	NESTED_TABLE
ALL_SUMDELTA	SEQUENCE
ALL_TAB_COL_STATISTICS	OWNER
ALL_TAB_COLUMNS	CHAR_LENGTH
	CHAR_USED
	V80_FMT_IMAGE
	DATA_UPGRADED
ALL_TAB_PRIVS	HIERARCHY
ALL_TAB_PRIVS_MADE	HIERARCHY
ALL_TAB_PRIVS_RECD	HIERARCHY
ALL_TABLES	DEPENDENCIES
ALL_TYPE_ATTRS	INHERITED
ALL_TYPE_METHODS	FINAL
	INSTANTIABLE
	OVERRIDING
	INHERITED
ALL_TYPES	FINAL
	INSTANTIABLE
	SUPERTYPE_OWNER
	SUPERTYPE_NAME
	LOCAL_ATTRIBUTES
	LOCAL_METHODS
	TYPEID
ALL_USTATS	PARTITION_NAME
ALL_VIEWS	SUPERVIEW_NAME

Static Data Dictionary View	New Columns
DBA_ALL_TABLES	DEPENDENCIES
DBA_ASSOCIATIONS	INTERFACE_VERSION
DBA_AUDIT_EXISTS	CLIENT_ID
	SESSION_CPU
DBA_AUDIT_OBJECT	CLIENT_ID
	SESSION_CPU
DBA_AUDIT_SESSION	CLIENT_ID
	SESSION_CPU
DBA_AUDIT_STATEMENT	CLIENT_ID
	SESSION_CPU
DBA_AUDIT_TRAIL	CLIENT_ID
	SESSION_CPU
DBA_CLUSTERS	DEPENDENCIES
DBA_CONSTRAINTS	INDEX_OWNER
	INDEX_NAME
	INVALID
	VIEW_RELATED
DBA_CONTEXT	TYPE
DBA_DIMENSIONS	COMPILE_STATE
DBA_IND_COLUMNS	CHAR_LENGTH
DBA_IND_PARTITIONS	DOMIDX_OPSTATUS
	PARAMETERS
DBA_INDEXES	JOIN_INDEX
DBA_INDEXTYPE_OPERATORS	OPERATOR_SCHEMA
DBA_INDEXTYPES	INTERFACE_VERSION
	PARTITIONING
DBA_MVIEWS	USE_NO_INDEX
DBA_OBJECT_TABLES	DEPENDENCIES
DBA_OPBINDINGS	PROPERTY

Static Data Dictionary View	New Columns
DBA_OUTLINES	SIGNATURE
DBA_PART_INDEXES	DEF_PARAMETERS
DBA_POLICIES	POLICY_GROUP
	STATIC_POLICY
DBA_REPCAT_TEMPLATE_OBJECTS	DDL_NUM
DBA_REPCAT_TEMPLATE_SITES	INSTANTIATION_DATE
DBA_REPCOLUMN	CTYPE_TOID
	CTYPE_OWNER
	CTYPE_HASHCODE
	CTYPE_MOD
	TOP
	CHAR_LENGTH
	CHAR_USED
DBA_REPDDL	DDL_NUM
DBA_REPFLAVOR_COLUMNS	TYPE_TOID
	TYPE_OWNER
	TYPE_HASHCODE
	TYPE_MOD
	TOP
DBA_REPOBJECT	NESTED_TABLE
DBA_RSRC_PLAN_DIRECTIVES	ACTIVE_SESS_POOL_P1
	QUEUEING_P1
	SWITCH_GROUP
	SWITCH_TIME
	SWITCH_ESTIMATE
	MAX_EST_EXEC_TIME
	UNDO_POOL
DBA_RSRC_PLANS	ACTIVE_SESS_POOL_MTH
	QUEUEING_MTH
DBA_TAB_COL_STATISTICS	OWNER

Static Data Dictionary View	New Columns
DBA_TAB_COLUMNS	CHAR_LENGTH
	CHAR_USED
	V80_FMT_IMAGE
	DATA_UPGRADED
DBA_TAB_PRIVS	HIERARCHY
DBA_TABLES	DEPENDENCIES
DBA_TABLESPACES	BLOCK_SIZE
	SEGMENT_SPACE_MANAGEMENT
DBA_TYPE_ATTRS	INHERITED
DBA_TYPE_METHODS	FINAL
	INSTANTIABLE
	OVERRIDING
	INHERITED
DBA_TYPES	FINAL
	INSTANTIABLE
	SUPERTYPE_OWNER
	SUPERTYPE_NAME
	LOCAL_ATTRIBUTES
	LOCAL_METHODS
	TYPEID
DBA_USTATS	PARTITION_NAME
DBA_VIEWS	SUPERVIEW_NAME
USER_ALL_TABLES	DEPENDENCIES
USER_ARGUMENTS	CHAR_LENGTH
	CHAR_USED
USER_ASSOCIATIONS	INTERFACE_VERSION
USER_AUDIT_OBJECT	CLIENT_ID
	SESSION_CPU
USER_AUDIT_SESSION	CLIENT_ID
	SESSION_CPU

Static Data Dictionary View	New Columns
USER_AUDIT_STATEMENT	CLIENT_ID
	SESSION_CPU
USER_AUDIT_TRAIL	CLIENT_ID
	SESSION_CPU
USER_CLUSTERS	DEPENDENCIES
USER_CONSTRAINTS	INDEX_OWNER
	INDEX_NAME
	INVALID
	VIEW_RELATED
USER_DIMENSIONS	COMPILE_STATE
USER_IND_COLUMNS	CHAR_LENGTH
USER_IND_PARTITIONS	DOMIDX_OPSTATUS
	PARAMETERS
USER_INDEXES	JOIN_INDEX
USER_INDEXTYPE_OPERATORS	OPERATOR_SCHEMA
USER_INDEXTYPES	INTERFACE_VERSION
	PARTITIONING
USER_MVIEWS	USE_NO_INDEX
USER_OBJECT_TABLES	DEPENDENCIES
USER_OPBINDINGS	PROPERTY
USER_OUTLINES	SIGNATURE
USER_PART_INDEXES	DEF_PARAMETERS
USER_POLICIES	POLICY_GROUP
	STATIC_POLICY
USER_REPCAT_TEMPLATE_OBJECTS	DDL_NUM
USER_REPCAT_TEMPLATE_SITES	INSTANTIATION_DATE

Static Data Dictionary View	New Columns
USER_REPCOLUMN	CTYPE_TOID
	CTYPE_OWNER
	CTYPE_HASHCODE
	CTYPE_MOD
	TOP
	CHAR_LENGTH
	CHAR_USED
USER_REPDDL	DDL_NUM
USER_REPFLAVOR_COLUMNS	TYPE_TOID
	TYPE_OWNER
	TYPE_HASHCODE
	TYPE_MOD
	TOP
USER_REPOBJECT	NESTED_TABLE
USER_REPPARAMETER_COLUMN	SNAME
USER_TAB_COLUMNS	CHAR_LENGTH
	CHAR_USED
	DATA_UPGRADED
	V80_FMT_IMAGE
USER_TAB_PRIVS	HIERARCHY
USER_TAB_PRIVS_MADE	HIERARCHY
USER_TAB_PRIVS_RECD	HIERARCHY
USER_TABLES	DEPENDENCIES
USER_TABLESPACES	BLOCK_SIZE
	SEGMENT_SPACE_MANAGEMENT
USER_TYPE_ATTRS	INHERITED
USER_TYPE_METHODS	FINAL
	INSTANTIABLE
	OVERRIDING
	INHERITED

Static Data Dictionary View	New Columns
USER_TYPES	FINAL
	INSTANTIABLE
	SUPERTYPE_OWNER
	SUPERTYPE_NAME
	LOCAL_ATTRIBUTES
	LOCAL_METHODS
	TYPEID
USER_USTATS	PARTITION_NAME
USER_VIEWS	SUPERVIEW_NAME

See Also: Chapter 2, "Static Data Dictionary 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$ARCHIVE and V$ARCHIVE	ISCURRENT

Dynamic Performance View	New Columns
GV$ARCHIVE_DEST and V$ARCHIVE_DEST	ARCHIVER
	SCHEDULE
	LOG_SEQUENCE
	DELAY_MINS
	PROCESS
	MANIFEST
	FAILURE_COUNT
	MAX_FAILURE
	ALTERNATE
	DEPENDENCY
	REGISTER
	QUOTA_SIZE
	QUOTA_USED
	MOUNTID
	TRANSMIT_MODE
	AFFIRM
	TYPE
GV$ARCHIVED_LOG and V$ARCHIVED_LOG	DEST_ID
	APPLIED
	STATUS
	DICTIONARY_BEGIN
	DICTIONARY_END
	END_OF_REDO
	BACKUP_COUNT
	ARCHIVAL_THREAD#
GV$BACKUP_SET and V$BACKUP_SET	KEEP
	KEEP_UNTIL
	KEEP_OPTIONS

Dynamic Performance View	New Columns
GV$BUFFER_POOL and V$BUFFER_POOL	BLOCK_SIZE
	RESIZE_STATE
	CURRENT_SIZE
	TARGET_SIZE
	TARGET_BUFFERS
	PREV_SIZE
	PREV_BUFFERS
GV$BUFFER_POOL_STATISTICS and V$BUFFER_POOL_STATISTICS	BLOCK_SIZE
GV$CIRCUIT and V$CIRCUIT	PCIRCUIT
GV$DATABASE and V$DATABASE	STANDBY_MODE
	REMOTE_ARCHIVE
	ACTIVATION#
	DATABASE_ROLE
	ARCHIVELOG_CHANGE#
	SWITCHOVER_STATUS
GV$DATAFILE and V$DATAFILE	BLOCK1_OFFSET
	AUX_NAME
GV$DATAFILE_COPYandV$DATAFILE_COPY	STATUS
	KEEP
	KEEP_UNTIL
	KEEP_OPTIONS
GV$DB_CACHE_ADVICE and V$DB_CACHE_ADVICE	BLOCK_SIZE
	ADVICE_STATUS
GV$DELETED_OBJECT and V$DELETED_OBJECT	OBJECT_DATA
GV$FILESTAT and V$FILESTAT	SINGLEBLKRDS
	SINGLEBLKRDTIM
GV$INSTANCE and V$INSTANCE	ACTIVE_STATE

Dynamic Performance View	New Columns
GV$INSTANCE_RECOVERY and V$INSTANCE_RECOVERY	TARGET_MTTR
	ESTIMATED_MTTR
	CKPT_BLOCK_WRITES
GV$LATCH and V$LATCH	WAIT_TIME
GV$LATCH_CHILDREN and V$LATCH_CHILDREN	WAIT_TIME
GV$LATCH_MISSES and V$LATCH_MISSES	LOCATION
GV$LATCH_PARENT and V$LATCH_PARENT	WAIT_TIME
GV$LOGFILE and V$LOGFILE	TYPE
GV$LOGMNR_CONTENTS and V$LOGMNR_CONTENTS	CSCN
	COMMIT_TIMESTAMP
	TX_NAME
	OPERATION_CODE
	SEQUENCE#
	REDO_VALUE
	UNDO_VALUE
	SQL_COLUMN_TYPE
	SQL_COLUMN_NAME
	REDO_LENGTH
	REDO_OFFSET
	UNDO_LENGTH
	UNDO_OFFSET
GV$LOGMNR_LOGS and V$LOGMNR_LOGS	DICTIONARY_BEGIN
	DICTIONARY_END
GV$MLS_PARAMETERS and V$MLS_PARAMETERS	UPDATE_COMMENT
GV$PARAMETER and V$PARAMETER	UPDATE_COMMENT
GV$PARAMETER2 and V$PARAMETER2	ORDINAL
	UPDATE_COMMENT

Dynamic Performance View	New Columns
GV$PROCESS and V$PROCESS	PGA_USED_MEM
	PGA_ALLOC_MEM
	PGA_MAX_MEM
GV$PROXY_DATAFILE and V$PROXY_DATAFILE	KEEP
	KEEP_UNTIL
	KEEP_OPTIONS
GV$RECOVER_FILE and V$RECOVER_FILE	ONLINE_STATUS
GV$RSRC_CONSUMER_GROUP and V$RSRC_CONSUMER_GROUP	QUEUE_LENGTH
	CURRENT_UNDO_CONSUMPTION
GV$SESSION and V$SESSION	CURRENT_QUEUE_DURATION
	CLIENT_IDENTIFIER
GV$SESSION_EVENTandV$SESSION_EVENT	TIME_WAITED_MICRO
GV$SORT_USAGE and V$SORT_USAGE	USERNAME
GV$SQL and V$SQL	PLAN_HASH_VALUE
	CPU_TIME
	ELAPSED_TIME
	OUTLINE_SID
	CHILD_ADDRESS
	SQLTYPE
	REMOTE
	OBJECT_STATUS
	LITERAL_HASH_VALUE
	LAST_LOAD_TIME
	IS_OBSOLETE
GV$SQL_SHARED_CURSOR and V$SQL_SHARED_CURSOR	LOGMINER_SESSION_MISMATCH
	INCOMP_LTRL_MISMATCH
	OVERLAP_TIME_MISMATCH
	USER_BIND_PEEK_MISMATCH
	TYPCHK_DEP_MISMATCH

Dynamic Performance View	New Columns
GV$SQLAREA and V$SQLAREA	CPU_TIME
	ELAPSED_TIME
	IS_OBSOLETE
GV$SYSTEM_EVENT and V$SYSTEM_EVENT	TIME_WAITED_MICRO
GV$SYSTEM_PARAMETER and V$SYSTEM_PARAMETER	UPDATE_COMMENT
GV$SYSTEM_PARAMETER2 and V$SYSTEM_PARAMETER2	ORDINAL
	UPDATE_COMMENT
GV$TABLESPACE and V$TABLESPACE	INCLUDED_IN_DATABASE_BACKUP
GV$TEMPSTAT and V$TEMPSTAT	SINGLEBLKRDS
	SINGLEBLKRDTIM
GV$TRANSACTION and V$TRANSACTION	NAME

See Also: Chapter 3, "Dynamic Performance (V$) Views" for more information about the new columns in these dynamic performance views

1

Initialization Parameters

This chapter contains detailed descriptions of the database initialization parameters and includes the following topics:

- Uses of Initialization Parameters
- Parameter Files
- Changing Parameter Values
- Reading the Parameter Descriptions
- 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.

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 the Initialization Parameter File

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

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

- The 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 1-65

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 the Initialization Parameter File

Character	Name	Description
#	Number sign	Comment
(	Left parenthesis	Start list of values
)	Right parenthesis	End list of values
"	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 the 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 Parameter Files" on page 1-5, 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

You change the value of a parameter by editing the initialization parameter file. In most cases, the new value takes effect the next time you start an instance of the

database. However, you can change the value of some parameters for the duration of the current session, as described in the following section.

Dynamic Parameters

Some initialization parameters are **dynamic**, that is, they can be modified using the ALTER SESSION or ALTER SYSTEM statements while an instance is running.

Use the following syntax to dynamically alter initialization parameters:

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

Whenever a dynamic parameter is modified using the ALTER SYSTEM statement, Oracle records the command 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:

CURSOR_SHARING	DB_BLOCK_CHECKING
DB_CREATE_FILE_DEST	DB_CREATE_ONLINE_LOG_DEST_*n*
DB_FILE_MULTIBLOCK_READ_COUNT	FILESYSTEMIO_OPTIONS
GLOBAL_NAMES	HASH_AREA_SIZE
HASH_JOIN_ENABLED	LOG_ARCHIVE_DEST_*n*
LOG_ARCHIVE_DEST_STATE_*n*	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_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_INDEX_CACHING	OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_MAX_PERMUTATIONS	OPTIMIZER_MODE
ORACLE_TRACE_ENABLE	PARALLEL_INSTANCE_GROUP
PARALLEL_MIN_PERCENT	PARTITION_VIEW_ENABLED
PLSQL_COMPILER_FLAGS	PLSQL_V2_COMPATIBILITY
QUERY_REWRITE_ENABLED	QUERY_REWRITE_INTEGRITY
REMOTE_DEPENDENCIES_MODE	SESSION_CACHED_CURSORS
SORT_AREA_RETAINED_SIZE	SORT_AREA_SIZE
STAR_TRANSFORMATION_ENABLED	STATISTICS_LEVEL
TIMED_OS_STATISTICS	TIMED_STATISTICS
TRACEFILE_IDENTIFIER	UNDO_SUPPRESS_ERRORS
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
BACKGROUND_DUMP_DEST	CONTROL_FILE_RECORD_KEEP_TIME
CORE_DUMP_DEST	CURSOR_SHARING
DB_*n*K_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_KEEP_CACHE_SIZE	DB_RECYCLE_CACHE_SIZE
DG_BROKER_CONFIG_FILE*n*	DG_BROKER_START
DISPATCHERS	DRS_START
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
JOB_QUEUE_PROCESSES	LARGE_POOL_SIZE
LICENSE_MAX_SESSIONS	LICENSE_MAX_USERS
LICENSE_SESSIONS_WARNING	LOCAL_LISTENER
LOG_ARCHIVE_DEST	LOG_ARCHIVE_DEST_*n*
LOG_ARCHIVE_DEST_STATE_*n*	LOG_ARCHIVE_DUPLEX_DEST
LOG_ARCHIVE_MAX_PROCESSES	LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_ARCHIVE_TRACE	LOG_CHECKPOINTS_TO_ALERT
LOG_CHECKPOINT_INTERVAL	LOG_CHECKPOINT_TIMEOUT
MAX_DUMP_FILE_SIZE	NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP	OPEN_CURSORS
OPTIMIZER_DYNAMIC_SAMPLING	ORACLE_TRACE_ENABLE
PARALLEL_ADAPTIVE_MULTI_USER	PARALLEL_INSTANCE_GROUP
PARALLEL_THREADS_PER_CPU	PGA_AGGREGATE_TARGET
PLSQL_COMPILER_FLAGS	PLSQL_NATIVE_C_COMPILER
PLSQL_NATIVE_LIBRARY_DIR	PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
PLSQL_NATIVE_LINKER	PLSQL_NATIVE_MAKE_FILE_NAME
PLSQL_NATIVE_MAKE_UTILITY	PLSQL_V2_COMPATIBILITY
QUERY_REWRITE_ENABLED	QUERY_REWRITE_INTEGRITY
REMOTE_DEPENDENCIES_MODE	REMOTE_LISTENER
RESOURCE_LIMIT	RESOURCE_MANAGER_PLAN
SERVICE_NAMES	SHARED_POOL_SIZE
SHARED_SERVERS	STANDBY_ARCHIVE_DEST
STANDBY_FILE_MANAGEMENT	STATISTICS_LEVEL
TIMED_OS_STATISTICS	TIMED_STATISTICS
TRACE_ENABLED	UNDO_RETENTION
UNDO_SUPPRESS_ERRORS	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:

BACKUP_TAPE_IO_SLAVES	OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE	OLAP_PAGE_POOL_SIZE
SORT_AREA_RETAINED_SIZE	SORT_AREA_SIZE
TRANSACTION_AUDITING	

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, 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 Corporation 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

Parameter type Specifies the type of the parameter:

- A Boolean parameter accepts either `true` or `false` 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.

Default value	Specifies the value this parameter assumes if not explicitly specified.
Parameter class	Specifies whether the parameter is dynamic or static. If dynamic, then it also specifies whether its value can be changed by an `ALTER SESSION` or `ALTER SYSTEM` statement. `ALTER SESSION` 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 `ALTER SESSION` statement. `ALTER SYSTEM` sets or changes the value of the parameter for all sessions.
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.

Parameter Descriptions

Descriptions of the individual initialization parameters follow 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:

- *Oracle9i Database Migration* for information about obsolete parameters
- Your system release bulletins or other operating system-specific Oracle documentation

ACTIVE_INSTANCE_COUNT

Parameter type	Integer
Default value	There is no default value.
Parameter class	Static
Range of values	1 or >= the number of instances in the cluster. (Values other than 1 have no effect on the active or standby status of any instances.)
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: *Oracle9i Real Application Clusters Setup and Configuration* for more information on setting this parameter

AQ_TM_PROCESSES

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM

Range of values 0 to 10

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:** *Oracle9i Application Developer's Guide - Advanced Queuing* for more information about this parameter and Advanced Queuing

ARCHIVE_LAG_TARGET

Parameter type	Integer
Default value	0 (disabled)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 or any integer in [60, 7200]
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 a user-specified time period 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:** *Oracle9i Data Guard Concepts and Administration*For more information about criteria for setting this parameter

AUDIT_FILE_DEST

Parameter type	String
Syntax	AUDIT_FILE_DEST = *'directory'*
Default value	*ORACLE_HOME*/rdbms/audit
Parameter class	Static

AUDIT_FILE_DEST specifies the directory where Oracle stores auditing files.

AUDIT_SYS_OPERATIONS

Parameter type	Boolean
Default value	false
Parameter class	Static
Range of values	true | false

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

Parameter type	String
Syntax	AUDIT_TRAIL = {NONE | FALSE | DB | TRUE | OS}
Default value	There is no default value.
Parameter class	Static

AUDIT_TRAIL enables or disables the automatic writing of rows to the audit trail.

Values:

- NONE or FALSE

Audit records are not written.

- OS

  Enables system-wide auditing and causes audited records to be written to the operating system's audit trail.

- DB or TRUE

  Enables system-wide auditing and causes audited records to be written to the database audit trail (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

Parameter type	String
Syntax	BACKGROUND_CORE_DUMP = {partial | full}
Default value	partial
Parameter class	Static

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 1-149

BACKGROUND_DUMP_DEST

Parameter type	String
Syntax	BACKGROUND_DUMP_DEST = {*pathname* | *directory*}

Default value	Operating system-dependent
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any valid local path, directory, or disk

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:

- *Oracle9i 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 1-169 for information on setting a destination for server process trace files

BACKUP_TAPE_IO_SLAVES

Parameter type	Boolean
Default value	`false`
Parameter class	Dynamic: `ALTER SYSTEM ... DEFERRED`
Range of values	`true | false`

BACKUP_TAPE_IO_SLAVES specifies whether I/O server processes (also called **slaves**) are used by the 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:

- *Oracle9i Recovery Manager User's Guide* for more information on duplexed backups
- "DBWR_IO_SLAVES" on page 1-46

BITMAP_MERGE_AREA_SIZE

Parameter type	Integer
Default value	`1048576` (1 MB)
Parameter class	Static
Range of values	Operating system-dependent

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: *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter

BLANK_TRIMMING

Parameter type Boolean

Default value `false`

Parameter class Static

Range of values `true | false`

`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.

BUFFER_POOL_KEEP

Parameter type String

Syntax `BUFFER_POOL_KEEP = {`*`integer`* `|`

`(BUFFERS:`*`integer`*`, LRU_LATCHES:`*`integer`*`)}`

where *`integer`* is the number of buffers and, optionally, the number of LRU latches.

Default value There is no default value.

Parameter class Static

Note: This parameter is deprecated in favor of the DB_KEEP_CACHE_SIZE parameter. Oracle recommends that you use DB_KEEP_CACHE_SIZE instead. Also, BUFFER_POOL_KEEP cannot be combined with the new dynamic DB_KEEP_CACHE_SIZE parameter; combining these parameters in the same parameter file will produce an error. BUFFER_POOL_KEEP is retained for backward compatibility only.

BUFFER_POOL_KEEP lets you save objects in the buffer cache by setting aside a portion of the total number of buffers (the value of the DB_BLOCK_BUFFERS parameter) as a KEEP buffer pool. You can also allocate to the KEEP buffer pool a specified portion of the total number of LRU latches.

The string value can take one of two forms. You can simply specify a buffer count:

```
BUFFER_POOL_KEEP = 5
```

Or, you can specify a combination of buffers and LRU latches. Specifying LRU latches is optional:

```
BUFFER_POOL_KEEP = (BUFFERS:400, LRU_LATCHES:3)
```

This parameter is most useful in combination with the BUFFER_POOL_RECYCLE parameter.

See Also:

- "BUFFER_POOL_RECYCLE" on page 1-22
- *Oracle9i Database Performance Tuning Guide and Reference* for information on setting these parameters and on using multiple buffer pools

BUFFER_POOL_RECYCLE

Parameter type String

Syntax BUFFER_POOL_RECYCLE = {*integer* |
(BUFFERS:*integer*, LRU_LATCHES:*integer*)}

where *integer* is the number of buffers and, optionally, the number of LRU latches.

Default value There is no default value.

Parameter class Static

Note: This parameter is deprecated in favor of the DB_RECYCLE_CACHE_SIZE parameter. Oracle recommends that you use DB_RECYCLE_CACHE_SIZE instead. Also, BUFFER_POOL_RECYCLE cannot be combined with the new dynamic DB_RECYCLE_CACHE_SIZE parameter; combining these parameters in the same parameter file will produce an error. BUFFER_POOL_RECYCLE is retained for backward compatibility only.

BUFFER_POOL_RECYCLE lets you limit the size of objects in the buffer cache by setting aside a portion of the total number of buffers (the value of the DB_BLOCK_BUFFERS parameter) as a RECYCLE buffer pool. You can also allocate to the RECYCLE buffer pool a specified portion of the total number of LRU latches.

The string value can take one of two forms. You can simply specify a buffer count:

```
BUFFER_POOL_RECYCLE = 5
```

Or, you can specify a combination of buffers and LRU latches. Specifying LRU latches is optional:

```
BUFFER_POOL_RECYCLE = (BUFFERS:50, LRU_LATCHES:1)
```

This parameter is most useful in combination with the BUFFER_POOL_KEEP parameter.

See Also:

- "BUFFER_POOL_KEEP" on page 1-21
- *Oracle9i Database Performance Tuning Guide and Reference* for information on setting these parameters and on using multiple buffer pools

CIRCUITS

Parameter type Integer

Default value Derived:

- If you are using shared server architecture, then the value of `SESSIONS`
- If you are not using the shared server architecture, then the value is `0`

Parameter class Static

`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.

See Also:

- *Oracle9i Database Concepts* for more information on memory structures and processes
- "DB_BLOCK_BUFFERS" on page 1-33 and "TRANSACTIONS" on page 1-164, which also contribute to SGA requirements

CLUSTER_DATABASE

Parameter type Boolean

Default value `false`

Parameter class Static

Range of values `true | false`

Real Application Clusters Multiple instances must have the same value.

`CLUSTER_DATABASE` is an Oracle9*i* Real Application Clusters parameter that specifies whether or not Oracle9*i* Real Application Clusters is enabled.

See Also: *Oracle9i Real Application Clusters Administration*

CLUSTER_DATABASE_INSTANCES

Parameter type	Integer
Default value	1
Parameter class	Static
Range of values	Any nonzero value

CLUSTER_DATABASE_INSTANCES is an Oracle9*i* 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 Oracle9*i* 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution
- *Oracle9i Real Application Clusters Administration* for information on Oracle9*i* Real Application Clusters

CLUSTER_INTERCONNECTS

Parameter type	String
Syntax	CLUSTER_INTERCONNECTS = *ifn* [: *ifn* ...]
Default value	There is no default value.
Parameter class	Static
Range of values	One or more IP addresses, separated by colons

CLUSTER_INTERCONNECTS provides Oracle with information about additional cluster interconnects available for use in Oracle9*i* 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 availibility 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: *Oracle9i Real Application Clusters Administration*

COMMIT_POINT_STRENGTH

Parameter type	Integer
Default value	1
Parameter class	Static
Range of values	0 to 255

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:

- *Oracle9i Database Concepts* and *Oracle9i Database Administrator's Guide* for information on two-phase commit
- Your operating system-specific Oracle documentation for the default value of this parameter

COMPATIBLE

Parameter type	String
Syntax	COMPATIBLE = *release_number*
Default value	8.1.0
Parameter class	Static
Range of values	Default release to current release
Real Application Clusters	Multiple instances must have the same value.

COMPATIBLE allows you to use a new release, 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 the Oracle server 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 current release may be restricted.

When using the standby database feature, this parameter must have the same value on the primary and standby databases, and the value must be 7.3.0.0.0 or higher. The default value is the earliest release with which compatibility can be guaranteed.

The value of COMPATIBLE must be set to 8.1.0 or higher in order to use stored columns of UROWID type. You can use ROWID pseudo columns for index-organized tables regardless of the parameter setting.

See Also: *Oracle9i Database Migration* for more information on setting this parameter

CONTROL_FILE_RECORD_KEEP_TIME

Parameter type	Integer
Default value	7 (days)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to 365 (days)

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
- *Oracle9i Database Concepts*
- *Oracle9i User-Managed Backup and Recovery Guide*

CONTROL_FILES

Parameter type	String
Syntax	CONTROL_FILES = *filename* [, *filename* [...]] **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
Parameter class	Static

Range of values	1 to 8 filenames
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 Corporation recommends that you multiplex multiple control files on different devices or mirror the file at the operating system level.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference*
- *Oracle9i Database Administrator's Guide*

CORE_DUMP_DEST

Parameter type	String
Syntax	CORE_DUMP_DEST = *directory*
Default value	*ORACLE_HOME*/DBS
Parameter class	Dynamic: ALTER SYSTEM

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

Parameter type	Integer
Default value	Set automatically by Oracle
Parameter class	Static
Range of values	0 to unlimited

> **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

Parameter type	Integer
Default value	8388608 (8 MB)
Parameter class	Static
Range of values	Operating system-dependent

> **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:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on using bitmap indexes

CURSOR_SHARING

Parameter type	String
Syntax	CURSOR_SHARING = {SIMILAR | EXACT | FORCE}
Default value	EXACT
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

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: *Oracle9i Database Performance Tuning Guide and Reference* for information on setting this parameter in these and other environments

CURSOR_SPACE_FOR_TIME

Parameter type	Boolean
Default value	false

Parameter class Static

Range of values `true | false`

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:

- *Oracle9i Database Concepts* and *Oracle9i Database Performance Tuning Guide and Reference*
- "SERIAL _REUSE" on page 1-145

DB_*n*K_CACHE_SIZE

Parameter type Big integer

Syntax `DB_[2 | 4 | 8 | 16 | 32]K_CACHE_SIZE = integer [K | M | G]`

Default value	0 (additional block size caches are not configured by default)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent

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.

Platform-specific block size restrictions apply. For example, you cannot set DB_32K_CACHE_SIZE if the maximum block size on the platform 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:** For more information on block size restrictions, see your platform specific documentation

DB_BLOCK_BUFFERS

Parameter type	Integer
Default value	Derived: 48 MB / DB_BLOCK_SIZE
Parameter class	Static
Range of values	50 to an operating system-specific maximum
Real Application Clusters	Multiple instances can have different values, and you can change the values as needed.

Note: This parameter is deprecated in favor of the DB_CACHE_SIZE parameter. Oracle recommends that you use DB_CACHE_SIZE instead. Also, DB_BLOCK_BUFFERS cannot be combined with the new 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.

The buffers for the "keep" buffer pool and the "recycle" buffer pool come out of the buffer cache. In addition, each LRU list in the buffer cache must have at least 50 buffers. Therefore, if you specify BUFFER_POOL_KEEP and BUFFER_POOL_RECYCLE, be sure the value of DB_BLOCK_BUFFERS conforms to the following formula:

```
DB_BLOCK_BUFFERS > #_buffers_in_"keep"_buffer_pool
                 + #_BUFFERS_IN_"recycle"_buffer_pool
                 + 50*(DB_BLOCK_LRU_LATCHES
                       - #_lru_latches_in_"keep"_buffer_pool
                       - #_lru_latches_in_"recycle"_buffer_pool)
```

If you did not specify any LRU latches in BUFFER_POOL_KEEP and BUFFER_POOL_RECYCLE, then the values default to 1.

This parameter affects the probability of data block pinging when Oracle9*i* Real Application Clusters are enabled: the more buffers, the greater the chance of pinging.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference*, "BUFFER_POOL_KEEP" on page 1-21, and "BUFFER_POOL_RECYCLE" on page 1-22 for more information on setting this parameter
- *Oracle9i Real Application Clusters Concepts* for more information on data block pinging
- Your operating system-specific Oracle documentation

DB_BLOCK_CHECKING

Parameter type	Boolean
Default value	`false`
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`

`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

Parameter type	Boolean
Default value	`true`
Parameter class	Dynamic: `ALTER SYSTEM`

Range of values `true | false`

DB_BLOCK_CHECKSUM determines whether DBW*n* 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`, DBW*n* 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 Corporation recommends that you set DB_BLOCK_CHECKSUM to `true`.

DB_BLOCK_SIZE

Parameter type	Integer
Default value	2048
Parameter class	Static
Range of values	2048 to 32768, but your operating system may have a narrower range
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 the size (in bytes) of Oracle database blocks. Typical values are 2048 and 4096. 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.

If you are using Oracle9*i* Real Application Clusters, then 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:

- *Oracle9i Database Performance Tuning Guide and Reference* and *Oracle9i Database Administrator's Guide* for information on setting this parameter
- *Oracle9i SQL Reference* for information on freelist groups
- *Oracle9i Database Performance Tuning Guide and Reference* for more information on the DSS and data warehouse environments
- *Oracle9i Database Concepts* for general information on block sizes

DB_CACHE_ADVICE

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
Parameter class	Dynamic: ALTER SYSTEM

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

Parameter type	Big integer
Syntax	DB_CACHE_SIZE = *integer* [K | M | G]
Default value	48 MB, rounded up to the nearest granule size
Parameter class	Dynamic: ALTER SYSTEM

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 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.

DB_CREATE_FILE_DEST

Parameter type	String
Syntax	DB_CREATE_FILE_DEST = *directory*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

DB_CREATE_FILE_DEST sets the default location for Oracle-managed datafiles. This location is also used as the default for Oracle-managed control files and online redo logs if DB_CREATE_ONLINE_LOG_DEST_*n* is not specified.

You can specify a file system directory as the default location for the creation of datafiles, control files, and online redo logs. However, 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.

DB_CREATE_ONLINE_LOG_DEST_*n*

Parameter type	String
Syntax	DB_CREATE_ONLINE_LOG_DEST_[1 | 2 | 3 | 4 | 5] = *directory*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

DB_CREATE_ONLINE_LOG_DEST_*n* (where *n* = 1, 2, 3, ... 5) sets the default location for Oracle-managed control files and online redo logs.

You should specify at least two parameters: DB_CREATE_ONLINE_LOG_DEST_1 and DB_CREATE_ONLINE_LOG_DEST_2. This provides greater fault tolerance for the logs if one of the destinations should fail.

If more than one directory is specified, then the control file or online redo log is multiplexed across the directories. One member of each online redo log is created in each directory, and one control file is created in each directory.

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.

DB_DOMAIN

Parameter type	String
Syntax	DB_DOMAIN = *domain_name*

Default value	There is no default value.
Parameter class	Static
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.
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 Corporation 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- The data dictionary view "GLOBAL_NAME" on page 2-220

DB_FILE_MULTIBLOCK_READ_COUNT

Parameter type	Integer
Default value	8
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

Range of values Operating system-dependent

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value of this parameter

DB_FILE_NAME_CONVERT

Parameter type String

Syntax

```
DB_FILE_NAME_CONVERT = [()'string1' ,
'string2' , 'string3' , 'string4' , ...[])]
```

Where:

- `string1` is the pattern of the primary database filename
- `string2` is the pattern of the standby database filename
- `string3` is the pattern of the primary database filename
- `string4` is the pattern of the standby database filename

You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks. The parentheses are optional.

Following are example settings that are acceptable:

```
DB_FILE_NAME_CONVERT =
('/dbs/t1/','/dbs/t1/s_','dbs/t2/
','dbs/t2/s_')
```

Default value None

Parameter class Static

`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 multivalue 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:

- *Oracle9i User-Managed Backup and Recovery Guide*
- *Oracle9i Data Guard Concepts and Administration*
- *Oracle9i Recovery Manager User's Guide*

DB_FILES

Parameter type	Integer
Default value	200
Parameter class	Static
Range of values	Minimum: the largest among the absolute file numbers of the datafiles in the database Maximum: operating system-dependent
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:

- *Oracle9i Real Application Clusters Administration* for information on setting this parameter in an Oracle9*i* Real Application Clusters environment
- Your operating system-specific Oracle documentation for the default value of this parameter

DB_KEEP_CACHE_SIZE

Parameter type	Big integer
Syntax	DB_KEEP_CACHE_SIZE = *integer* [K | M | G]
Default value	0 (KEEP cache is not configured by default)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent

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 parameter).

DB_NAME

Parameter type	String
Syntax	DB_NAME = *database_name*
Default value	There is no default value.
Parameter class	Static
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 STARTUP OPEN SQL*Plus statement or the ALTER DATABASE MOUNT SQL statement.

DB_NAME specifies a database identifier of up to 8 characters. If specified, it must correspond to the name specified in the CREATE DATABASE statement. Although the use of DB_NAME is optional, you should generally set it before issuing the CREATE DATABASE statement, and then reference it in that 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.

If you do not specify this parameter, you must specify a database name in either the STARTUP command or the ALTER DATABASE ... MOUNT statement for each instance of the cluster database.

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:** *Oracle9i Database Administrator's Guide* and *Oracle9i Real Application Clusters Administration* for more information on setting this parameter

DB_RECYCLE_CACHE_SIZE

Parameter type	Big integer
Syntax	DB_RECYCLE_CACHE_SIZE = *integer* [K | M | G]
Default value	0 (RECYCLE cache is not configured by default)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent

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 parameter).

DB_WRITER_PROCESSES

Parameter type	Integer
Default value	1
Parameter class	Static
Range of values	1 to 20

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:

- *Oracle9i Database Concepts* for more information on the database writer processes
- *Oracle9i Database Performance Tuning Guide and Reference* for information on setting the DBW*n* parameters

DBLINK_ENCRYPT_LOGIN

Parameter type	Boolean
Default value	`false`
Parameter class	Static
Range of values	`true | false`

When you attempt to connect to a database using a password, Oracle encrypts the password before sending it to the database. DBLINK_ENCRYPT_LOGIN specifies whether or not attempts to connect to other Oracle servers through database links should use encrypted passwords.

Values:

- TRUE

  When a connection fails Oracle does not reattempt the connection.

- FALSE

  When a connection fails Oracle reattempts the connection using an unencrypted version of the password.

DBWR_IO_SLAVES

Parameter type	Integer
Default value	`0`

Parameter class	Static
Range of values	0 to operating system-dependent

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 1-19

DG_BROKER_CONFIG_FILE*n*

Parameter type	String
Syntax	DG_BROKER_CONFIG_FILE[1 | 2] = *filename*
Default value	Operating system-dependent
Parameter class	Dynamic: ALTER SYSTEM
Range of values	One filename

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:** *Oracle9i Data Guard Broker* for more information about setting this parameter

DG_BROKER_START

Parameter type	Boolean
Default value	false
Parameter class	Dynamic: ALTER SYSTEM
Range of values	true | false

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_ASYNCH_IO

Parameter type	Boolean
Default value	true
Parameter class	Static
Range of values	true | false

DISK_ASYNCH_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 Corporation 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_ASYNCH_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 1-46
- *Oracle9i Database Performance Tuning Guide and Reference* for more information on asynchronous I/O and on setting this parameter

DISPATCHERS

Parameter type String

Syntax

```
DISPATCHERS = 'dispatch_clause'
```

dispatch_clause::=

```
(PROTOCOL = protocol) |
(ADDRESS = address) |
(DESCRIPTION = description )
[options_clause]
```

options_clause::=

```
(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)
```

Default value There is no default value.

Parameter class Dynamic: ALTER SYSTEM

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)"
```

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 (PRO or PROT)

  The network protocol for which the dispatcher generates a listening endpoint.

- ADDRESS (ADD or ADDR)

  The network protocol address of the endpoint on which the dispatchers listen.

- DESCRIPTION (DES or DESC)

The network description of the endpoint on which the dispatchers listen, including the protocol address.

options_clause

- DISPATCHERS (DIS or DISP)

  The initial number of dispatchers to start. The default is 1.

- SESSIONS (SES or SESS)

  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 (CON or CONN)

  The maximum number of network connections to allow for each dispatcher. The default is operating system-specific.

- TICKS (TIC or TICK)

  The length of a network tick in seconds. The default is 15 seconds.

- POOL (POO)

  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 0 or 1, then the default value of 10 ticks is used.

- MULTIPLEX (MUL or MULT)

  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 (LIS, LIST)

  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 1-73 and "REMOTE_LISTENER" on page 1-138.

- SERVICE (SER, SERV)

  The service names the dispatchers register with the listeners.

- INDEX

  Use this parameter in an ALTER SYSTEM SET DISPATCHERS statement to indicate which dispatcher you want to modify. (If you specify INDEX in the initialization parameter file, Oracle 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 process) to one less than the total number of dispatchers you define.

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

See Also: *Oracle9i Net Services Administrator's Guide* for more information on setting this parameter

DISTRIBUTED_LOCK_TIMEOUT

Parameter type	Integer
Default value	60
Parameter class	Static
Range of values	1 to unlimited

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

See Also: *Oracle9i Database Concepts* and *Oracle9i Database Administrator's Guide* for more information on data concurrency

DML_LOCKS

Parameter type	Integer
Default value	Derived: 4 * TRANSACTIONS
Parameter class	Static
Range of values	20 to unlimited; a setting of 0 disables enqueues
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:

- *Oracle9i Database Concepts* for a discussion of lock and enqueue resources needed for parallel DML
- *Oracle9i Real Application Clusters Administration*, *Oracle9i Database Concepts*, and *Oracle9i Database Administrator's Guide* for more information on data concurrency

DRS_START

Parameter type	Boolean
Default value	false
Parameter class	Dynamic: ALTER SYSTEM
Range of values	true | false

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

DRS_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 `DRS_START` to `true` in the initialization parameter file.

ENQUEUE_RESOURCES

Parameter type	Integer
Default value	Derived from `SESSIONS` parameter
Parameter class	Static
Range of values	10 to unlimited

`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:

- *Oracle9i Real Application Clusters Administration*
- *Oracle9i Database Concepts*
- *Oracle9i Database Administrator's Guide*

EVENT

Parameter type	String
Default value	There is no default value.
Parameter class	Static

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

Parameter type	String
Syntax	FAL_CLIENT = *string*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SYSTEM

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: *Oracle9i Data Guard Concepts and Administration* for more information about FAL server

FAL_SERVER

Parameter type	String
Syntax	FAL_SERVER = *string*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SYSTEM

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:** *Oracle9i Data Guard Concepts and Administration* for more information about FAL server

FAST_START_IO_TARGET

Parameter type	Integer
Default value	All the buffers in the cache
Parameter class	Dynamic: ALTER SYSTEM
Range of values	1000 to all buffers in the cache. A setting of 0 disables limiting recovery I/Os.
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_MTTR_TARGET parameter. Oracle recommends that you use FAST_START_MTTR_TARGET instead. FAST_START_IO_TARGETis 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, DBW*n* 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, DBW*n* 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- V$INSTANCE_RECOVERY on page 3-68 for information on fast-start checkpointing

FAST_START_MTTR_TARGET

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to 3600 seconds
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

Parameter type	String
Syntax	FAST_START_PARALLEL_ROLLBACK = {HI | LO | FALSE}
Default value	LOW
Parameter class	Dynamic: ALTER SYSTEM

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

Parameter type	Boolean
Default value	false
Parameter class	Dynamic: ALTER SYSTEM
Range of values	true | false

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

FILESYSTEMIO_OPTIONS

Parameter type	String
Syntax	`FILESYSTEMIO_OPTIONS = {none | setall | directIO | asynch}`
Default value	There is no default value.
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM`

`FILESYSTEMIO_OPTIONS` specifies I/O operations for file system files.

FIXED_DATE

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.
Parameter class	Dynamic: `ALTER SYSTEM`

`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.

Specify the value either without quotation marks or with double quotation marks.

Note: Do not use single quotation marks.

GC_FILES_TO_LOCKS

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.
Parameter class	Static
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 Oracle9*i* Real Application Clusters.

GC_FILES_TO_LOCKS is an Oracle9*i* 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 FILE_NAME and FILE_ID columns of the "DBA_DATA_FILES" on page 2-159 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:** *Oracle9i Real Application Clusters Concepts* and *Oracle9i Real Application Clusters Deployment and Performance* for more information on setting this parameter

GLOBAL_CONTEXT_POOL_SIZE

Parameter type	String
Default value	1 MB
Parameter class	Static
Range of values	Any integer value in MB

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

GLOBAL_NAMES

Parameter type	Boolean
Default value	false

Parameter class Dynamic: ALTER SESSION, ALTER SYSTEM

Range of values true | false

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 Corporation 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: *Oracle9i Database Administrator's Guide* for more information on setting this parameter

HASH_AREA_SIZE

Parameter type Integer

Default value Derived: 2 * SORT_AREA_SIZE

Parameter class Dynamic: ALTER SESSION

Range of values 0 to operating system-dependent

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:

- *Oracle9i Database Concepts* for information on hash joins in general
- *Oracle9i Database Performance Tuning Guide and Reference* for information on calculating an appropriate value for this parameter

HASH_JOIN_ENABLED

Parameter type	Boolean
Default value	`true`
Parameter class	Dynamic: `ALTER SESSION`
Range of values	`true | false`

`HASH_JOIN_ENABLED` specifies whether the optimizer should consider using a hash join as a join method. If set to `false`, then hashing is not available as a join method. If set to `true`, then the optimizer compares the cost of a hash join with other types of joins, and chooses hashing if it gives the best cost. Oracle Corporation recommends that you set this parameter to `true` for all data warehousing applications.

HI_SHARED_MEMORY_ADDRESS

Parameter type	Integer
Default value	`0`
Parameter class	Static

`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 1-149). If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

HS_AUTOREGISTER

Parameter type	Boolean

Default value	`true`
Parameter class	Dynamic: `ALTER SYSTEM`
Range of values	`true | false`

`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 Corporation 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:** *Oracle9i Heterogeneous Connectivity Administrator's Guide* for more information on HS agents

IFILE

Parameter type	Parameter file
Syntax	`IFILE = parameter_file_name`
Default value	There is no default value.
Parameter class	Static
Range of values	Valid parameter filenames
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

Parameter type	String
Syntax	INSTANCE_GROUPS = *group_name* [, *group_name* ...]
Default value	There is no default value.
Parameter class	Static
Range of values	One or more instance group names, separated by commas
Real Application Clusters	Multiple instances can have different values.

INSTANCE_GROUPS is an Oracle9*i* 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:

- *Oracle9i Real Application Clusters Administration* for more information on parallel query execution in an Oracle9*i* Real Application Clusters environment
- "PARALLEL_INSTANCE_GROUP" on page 1-122

INSTANCE_NAME

Parameter type	String
Syntax	`INSTANCE_NAME = instance_id`
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.
Parameter class	Static
Range of values	Any alphanumeric characters

In an Oracle9*i* 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:** *Oracle9i Real Application Clusters Administration* and *Oracle9i Net Services Administrator's Guide* for more information

INSTANCE_NUMBER

Parameter type	Integer
Default value	Lowest available number; derived from instance start up order and `INSTANCE_NUMBER` value of other instances. If not configured for Oracle9*i* Real Application Clusters, then `0`.
Parameter class	Static
Range of values	1 to maximum number of instances specified when the database was created
Real Application Clusters	You must set this parameter for every instance, and all instances must have different values.

INSTANCE_NUMBER is an Oracle9*i* 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:** *Oracle9i Real Application Clusters Administration* for more information

JAVA_MAX_SESSIONSPACE_SIZE

Parameter type	Integer
Default value	0
Parameter class	Static
Range of values	0 to 4 GB

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:**
>
> - *Oracle9i Java Developer's Guide*
> - "JAVA_SOFT_SESSIONSPACE_LIMIT" on page 1-69

JAVA_POOL_SIZE

Parameter type	Big integer
Syntax	LARGE_POOL_SIZE = *integer* [K | M | G]

Default value	24 MB, rounded up to the nearest granule size
Parameter class	Static
Range of values	Minimum: the granule size Maximum: operating system-dependent

JAVA_POOL_SIZE specifies the size (in bytes) 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:** *Oracle9i Java Developer's Guide* for information on adjusting this parameter

JAVA_SOFT_SESSIONSPACE_LIMIT

Parameter type	Integer
Default value	0
Parameter class	Static
Range of values	0 to 4 GB

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:**
>
> - *Oracle9i Java Developer's Guide* for more information on this parameter
> - "JAVA_MAX_SESSIONSPACE_SIZE" on page 1-68

JOB_QUEUE_PROCESSES

Parameter type	Integer
Default value	0

Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to 1000
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:

- *Oracle9i Replication* and *Oracle9i Data Warehousing Guide* for more information on managing materialized views
- *Oracle9i Application Developer's Guide - Advanced Queuing* for more information on message propagation

LARGE_POOL_SIZE

Parameter type	Big integer
Syntax	LARGE_POOL_SIZE = *integer* [K | M | G]
Default value	0 if both of the following are true: ■ The pool is not required by parallel execution ■ DBWR_IO_SLAVES is not set Otherwise, derived from the values of PARALLEL_MAX_SERVERS, PARALLEL_THREADS_PER_CPU, CLUSTER_DATABASE_INSTANCES, DISPATCHERS, and DBWR_IO_SLAVES.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	300 KB to at least 2 GB (actual maximum is operating system-specific)

LARGE_POOL_SIZE lets you specify the size (in bytes) 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: *Oracle9i Database Performance Tuning Guide and Reference* and *Oracle9i Database Migration* for more information on setting this parameter

LICENSE_MAX_SESSIONS

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to number of session licenses
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 1-72).

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

> **See Also:** *Oracle9i Database Administrator's Guide* for more information on setting this parameter

LICENSE_MAX_USERS

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to number of user licenses
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:** *Oracle9i Database Administrator's Guide* for more information on setting this parameter

LICENSE_SESSIONS_WARNING

Parameter type	Integer
Default value	0

Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to value of LICENSE_MAX_SESSIONS parameter
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 1-71).

See Also:

- *Oracle9i Database Administrator's Guide* for more information on setting this parameter
- *Oracle9i Real Application Clusters Administration* for more information on calculating an appropriate value for this parameter

LOCAL_LISTENER

Parameter type	String
Syntax	LOCAL_LISTENER = *network_name*
Default value	(ADDRESS = (PROTOCOL=TCP)(HOST=)(PORT=1521))
Parameter class	Dynamic: ALTER SYSTEM

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:

- *Oracle9i Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle9i 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

Parameter type	String
Syntax	LOCK_NAME_SPACE = *namespace*
Default value	There is no default value.
Parameter class	Static
Range of values	Up to 8 alphanumeric characters. No special characters allowed.

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, set LOCK_NAME_SPACE in the standby initialization parameter file to a distinct value such as the following:

```
LOCK_NAME_SPACE = standby
```

LOCK_SGA

Parameter type	Boolean
Default value	false
Parameter class	Static
Range of values	true | false

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_DEST

Parameter type	String
Syntax	LOG_ARCHIVE_DEST = *filespec*
Default value	Null
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any valid path or device name, except raw partitions
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_*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_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_*n* parameters, and must be defined as the null string ("") or (' ') when any LOG_ARCHIVE_DEST_*n* 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_*n* 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:

- *Oracle9i User-Managed Backup and Recovery Guide*
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-79, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-82, and "V$ARCHIVE_DEST" on page 3-5 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*

Parameter type String

Syntax

```
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 ]
   }
```

Default value There is no default value.

Parameter class Dynamic: ALTER SESSION, ALTER SYSTEM

Note: This parameter is valid only if you have installed Oracle Enterprise Edition. You may continue to use LOG_ARCHIVE_DEST if you have installed 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_*n* parameters (where *n* = 1, 2, 3, ... 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 archivelog. 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, Oracle Corporation 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*, the default value is 300 seconds.

  See Also: *Oracle9i User-Managed Backup and Recovery Guide* and *Oracle9i Database Administrator's Guide* for more information and examples

LOG_ARCHIVE_DEST_STATE_*n*

Parameter type	String
Syntax	LOG_ARCHIVE_DEST_STATE_*n* = {alternate | reset | defer | enable}
Default value	enable
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

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:

- *Oracle9i User-Managed Backup and Recovery Guide* and *Oracle9i Database Administrator's Guide* for more information and examples
- "V$ARCHIVE_DEST" on page 3-5

LOG_ARCHIVE_DUPLEX_DEST

Parameter type	String
Syntax	LOG_ARCHIVE_DUPLEX_DEST = *filespec*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Either a null string or any valid path or device name, except raw partitions

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 1-76
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-82
- "V$ARCHIVE_DEST" on page 3-5

LOG_ARCHIVE_FORMAT

Parameter type	String
Syntax	LOG_ARCHIVE_FORMAT = *filename*
Default value	Operating system-dependent
Parameter class	Static
Range of values	Any string that resolves to a valid filename
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

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%s_%t.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:

- *Oracle9i User-Managed Backup and Recovery Guide* and *Oracle9i Real Application Clusters Administration* 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_MAX_PROCESSES

Parameter type	Integer
Default value	1
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any integer from 1 to 10

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: *Oracle9i Database Concepts*

LOG_ARCHIVE_MIN_SUCCEED_DEST

Parameter type	Integer
Default value	1
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM
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

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:

- *Oracle9i Database Administrator's Guide* for more information on setting this parameter
- "LOG_ARCHIVE_DEST_n" on page 1-76, "LOG_ARCHIVE_DUPLEX_DEST" on page 1-79, and "V$ARCHIVE_DEST" on page 3-5 for information on related parameters

LOG_ARCHIVE_START

Parameter type	Boolean
Default value	false
Parameter class	Static
Range of values	true | false
Real Application Clusters	Multiple instances can have different values.

LOG_ARCHIVE_START is applicable only when you use the redo log in ARCHIVELOG mode. It indicates whether archiving should be automatic or manual when the instance starts up.

Values:

- true

  Indicates that archiving is automatic

- false

  Indicates that the database administrator will archive filled redo log files manually

The SQL*Plus statements ARCHIVE LOG START or ARCHIVE LOG STOP override this parameter.

In ARCHIVELOG mode, if all online redo log files fill without being archived, Oracle issues an error and suspends instance operations until the necessary archiving is performed. This delay is more likely if you use manual archiving. You can reduce its likelihood by increasing the number of online redo log files.

To use ARCHIVELOG mode while creating a database, set this parameter to true. Normally, a database is created in NOARCHIVELOG mode and then altered to ARCHIVELOG mode after creation.

> **See Also:** *Oracle9i Database Administrator's Guide* and *Oracle9i Real Application Clusters Administration*

LOG_ARCHIVE_TRACE

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0, 1, 2, 4, 8, 16, 32, 64, 128
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: *Oracle9i User-Managed Backup and Recovery Guide*

LOG_BUFFER

Parameter type	Integer
Default value	512 KB or 128 KB * CPU_COUNT, whichever is greater
Parameter class	Static
Range of values	Operating system-dependent

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

LOG_CHECKPOINT_INTERVAL

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Unlimited
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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- "FAST_START_IO_TARGET" on page 1-57, "LOG_CHECKPOINT_TIMEOUT" on page 1-87, and "V$INSTANCE_RECOVERY" on page 3-68

LOG_CHECKPOINT_TIMEOUT

Parameter type	Integer
Default value	1800
Parameter class	Dynamic: ALTER SYSTEM
Range of values	0 to unlimited
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.

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- "FAST_START_IO_TARGET" on page 1-57, "LOG_CHECKPOINT_TIMEOUT" on page 1-87, and "V$INSTANCE_RECOVERY" on page 3-68

LOG_CHECKPOINTS_TO_ALERT

Parameter type	Boolean
Default value	`false`
Parameter class	Dynamic: `ALTER SYSTEM`
Range of values	`true | false`

`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

Parameter type	String

Syntax

```
LOG_FILE_NAME_CONVERT = [()'string1' ,
'string2' , 'string3' , 'string4' , ...[])]
```

Where:

- `string1` is the pattern of the primary database filename
- `string2` is the pattern of the standby database filename
- `string3` is the pattern of the primary database filename
- `string4` is the pattern of the standby database filename

You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks. The parentheses are optional.

Following are example settings that are acceptable:

```
LOG_FILE_NAME_CONVERT=('/dbs/t1/','/dbs/t1/s_
','dbs/t2/ ','dbs/t2/s_')
```

Default value None

Parameter class Static

Range of values Character strings

`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 multivalue 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:** *Oracle9i User-Managed Backup and Recovery Guide* and *Oracle9i Data Guard Concepts and Administration*

LOG_PARALLELISM

Parameter type Integer

Default value 1

Parameter class Static

Range of values 1 to 255

LOG_PARALLELISM specifies the level of concurrency for redo allocation within Oracle.

If you are using Oracle on high-end servers that have more than 16 processors, and you are experiencing very high contention on the redo allocation latch, then you should consider enabling parallel redo. This allows the parallel generation of redo and can increase the throughput of certain update-intensive workloads. You can examine the V$LATCH view to observe the cumulative wait times on the redo allocation latch.

To enable parallel redo, set the LOG_PARALLELISM parameter in the initialization parameter file to the desired level of concurrency for redo allocation. Oracle Corporation recommends setting LOG_PARALLELISM to a value between 2 and 8 when running on systems with 16 to 64 processors. You can start with a low value (such as 2) and increment it by 1 until you determine that the redo allocation latch is no longer contended, and that the parameter improves the performance of the application.

Setting LOG_PARALLELISM to values greater than 8 is not currently recommended.

LOGMNR_MAX_PERSISTENT_SESSIONS

Parameter type Integer

Default value 1

Parameter class Static

Range of values 1 to LICENSE_MAX_SESSIONS

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*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:** *Oracle9i Supplied PL/SQL Packages and Types Reference* for information on the DBMS_LOGMNR_SESSION package

MAX_COMMIT_PROPAGATION_DELAY

Parameter type Integer

Default value 700

Parameter class Static

Range of values 0 to 90000

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 an Oracle9*i* 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: *Oracle9i Real Application Clusters Administration* for more information on setting this parameter

MAX_DISPATCHERS

Parameter type	Integer
Default value	5
Parameter class	Static
Range of values	5 or the number of dispatchers configured, whichever is greater

MAX_DISPATCHERS specifies the maximum number of dispatcher processes allowed to be running simultaneously. The default value applies only if dispatchers have been configured for the system.

The value of MAX_DISPATCHERS should at least equal the maximum number of concurrent sessions divided by the number of connections for each dispatcher. For

most systems, a value of 250 connections for each dispatcher provides good performance.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* 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

Parameter type	String
Syntax	MAX_DUMP_FILE_SIZE = {*integer* [K | M] | UNLIMITED}
Default value	UNLIMITED
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM
Range of values	0 to unlimited, or UNLIMITED

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: *Oracle9i Database Administrator's Guide* and *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter

MAX_ENABLED_ROLES

Parameter type	Integer
Default value	20
Parameter class	Static
Range of values	0 to 148

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 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, user HR can have seven roles enabled: the five enabled by MAX_ENABLED_ROLES, plus PUBLIC and HR.

> **See Also:** *Oracle9i Database Administrator's Guide* for more information on setting this parameter

MAX_ROLLBACK_SEGMENTS

Parameter type	Integer
Default value	MAX(30, TRANSACTIONS/TRANSACTIONS_PER_ROLLBACK_SEGMENT)
Parameter class	Static
Range of values	2 to 65535

MAX_ROLLBACK_SEGMENTS specifies the maximum size of the rollback segment cache in the SGA. The number specified signifies the maximum number of rollback segments that can be kept online (that is, status of ONLINE) simultaneously by one instance.

> **See Also:** *Oracle9i Database Administrator's Guide* for more information on setting this parameter

MAX_SHARED_SERVERS

Parameter type	Integer

Default value	Derived from SHARED_SERVERS (either 20 or 2*SHARED_SERVERS)
Parameter class	Static
Range of values	Operating system-dependent

MAX_SHARED_SERVERS specifies the maximum number of shared server processes allowed to be running simultaneously. If artificial deadlocks occur too frequently on your system, you should increase the value of MAX_SHARED_SERVERS.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- *Oracle9i Database Concepts* for information on artificial deadlocks
- Your operating system-specific Oracle documentation for the default value and range of values

NLS_CALENDAR

Parameter type	String
Syntax	NLS_CALENDAR = "*calendar_system*"
Default value	None
Parameter class	Dynamic: ALTER SESSION
Range of values	Any valid calendar format name

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:** *Oracle9i Database Globalization Support Guide* for a listing of available calendar systems

NLS_COMP

Parameter type	String
Syntax	NLS_COMP = {BINARY | ANSI}
Default value	BINARY
Parameter class	Dynamic: ALTER SESSION

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:** *Oracle9i Database Globalization Support Guide* for more information on setting this parameter

NLS_CURRENCY

Parameter type	String
Syntax	NLS_CURRENCY = *currency_symbol*
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: ALTER SESSION
Range of values	Any valid character string, with a maximum of 10 bytes (not including null)

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:

- *Oracle9i Database Globalization Support Guide* for more information on setting this parameter
- *Oracle9i SQL Reference* for information on number format elements

NLS_DATE_FORMAT

Parameter type	String
Syntax	NLS_DATE_FORMAT = "*format*"
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: ALTER SESSION
Range of values	Any valid date format mask but not exceeding a fixed length

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:

- *Oracle9i Database Globalization Support Guide* for more information on setting this parameter
- *Oracle9i Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DATE_LANGUAGE

Parameter type	String
Syntax	NLS_DATE_LANGUAGE = *language*
Default value	Derived from NLS_LANGUAGE
Parameter class	Dynamic: ALTER SESSION
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:

- *Oracle9i Database Globalization Support Guide* for more information on setting this parameter
- *Oracle9i SQL Reference* for information on the TO_DATE and TO_CHAR functions.
- *Oracle9i Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DUAL_CURRENCY

Parameter type	String
Syntax	NLS_DUAL_CURRENCY = *currency_symbol*
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: 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: *Oracle9i Database Globalization Support Guide* for more information on setting this parameter

NLS_ISO_CURRENCY

Parameter type	String
Syntax	NLS_ISO_CURRENCY = *territory*
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic : 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:

- *Oracle9i Database Globalization Support Guide* for more information on setting this parameter
- *Oracle9i SQL Reference* for information on number format elements

NLS_LANGUAGE

Parameter type	String
Syntax	NLS_LANGUAGE = *language*
Default value	Operating system-dependent, derived from the NLS_LANG environment variable
Parameter class	Dynamic: ALTER SESSION
Range of values	Any valid language name

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:

- *Oracle9i 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

Parameter type	String
Syntax	NLS_LENGTH_SEMANTICS = *string*
	Example: NLS_LENGTH_SEMANTICS = 'CHAR'
Default value	BYTE
Parameter class	Dynamic: 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

Parameter type	String
Syntax	NLS_NCHAR_CONV_EXCP = {TRUE | FALSE}
Default value	FALSE
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

NLS_NCHAR_CONV_EXCP determines whether data loss during an implicit or explicit character type conversion will report an error.

See Also: *Oracle9i Database Globalization Support Guide* for more information on setting this parameter

NLS_NUMERIC_CHARACTERS

Parameter type	String
Syntax	NLS_NUMERIC_CHARACTERS = "*decimal_character group_separator*"
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: 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: *Oracle9i Database Globalization Support Guide* for more information on setting this parameter

NLS_SORT

Parameter type	String
Syntax	NLS_SORT = {BINARY | *linguistic_definition*}

Default value	Derived from NLS_LANGUAGE
Parameter class	Dynamic: 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:

- *Oracle9i Database Globalization Support Guide* for a current listing of values you can specify for this parameter
- *Oracle9i Database Concepts* for more information on this parameter

NLS_TERRITORY

Parameter type	String
Syntax	NLS_TERRITORY = *territory*
Default value	Operating system-dependent

Parameter class	Dynamic: ALTER SESSION
Range of values	Any valid territory name

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 1-97, "NLS_NUMERIC_CHARACTERS" on page 1-101, "NLS_CURRENCY" on page 1-96, and "NLS_ISO_CURRENCY" on page 1-99.

See Also:

- *Oracle9i 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

Parameter type	String
Syntax	NLS_TIMESTAMP_FORMAT = "*format*"
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: 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

Parameter type	String
Syntax	NLS_TIMESTAMP_TZ_FORMAT = "*format*"
Default value	Derived from NLS_TERRITORY
Parameter class	Dynamic: 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_TZfunctions.

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.

O7_DICTIONARY_ACCESSIBILITY

Parameter type	Boolean
Default value	false
Parameter class	Static
Range of values	true | false

O7_DICTIONARY_ACCESSIBILITY is intended for use when you migrate from Oracle7 to Oracle9*i*. It 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 SYS schema.

For example, if O7_DICTIONARY_ACCESSIBILITY = 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 privilege. Also, the following roles, which can be granted to the database administrator, also allow access to dictionary objects: SELECT_CATALOG_ROLE, EXECUTE_CATALOG_ROLE, and DELETE_CATALOG_ROLE.

See Also:

- *Oracle9i Database Migration* for more information on this parameter and the roles mentioned here
- *Oracle9i SQL Reference* for information on granting roles

OBJECT_CACHE_MAX_SIZE_PERCENT

Parameter type	Integer
Default value	10
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM ... DEFERRED
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 1-106 for a description of the object cache
- *Oracle9i Database Concepts*, *Pro*C/C++ Precompiler Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OBJECT_CACHE_OPTIMAL_SIZE

Parameter type	Integer
Default value	`102400` (100K)
Parameter class	Dynamic: `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: *Oracle9i Database Concepts*, *Pro*C/C++ Precompiler Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OLAP_PAGE_POOL_SIZE

Parameter type	Integer
Default value	32 MB
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM ... DEFERRED`
Range of values	32 MB to 2 GB

`OLAP_PAGE_POOL_SIZE` specifies the size (in bytes) of the OLAP pool.

OPEN_CURSORS

Parameter type	Integer
Default value	50
Parameter class	Static
Range of values	1 to 4294967295 (4 GB -1)

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. This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are reexecuted by a user.

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the range of values

OPEN_LINKS

Parameter type	Integer
Default value	4
Parameter class	Static
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 1-108 for information on setting open connections globally for a database instance

OPEN_LINKS_PER_INSTANCE

Parameter type	Integer
Default value	4
Parameter class	Static
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 1-107

OPTIMIZER_DYNAMIC_SAMPLING

Parameter type	Integer
Default value	If OPTIMIZER_FEATURES_ENABLE is set to 9.2.0 or higher, then 1 If OPTIMIZER_FEATURES_ENABLE is set to 9.0.1 or lower, then 0
Parameter class	Dynamic: 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: *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter

OPTIMIZER_FEATURES_ENABLE

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}
Default value	9.2.0
Parameter class	Static

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:** *Oracle9i Database Performance Tuning Guide and Reference* 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
Setting of optimizer mode for user recursive SQL											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
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

Table 1–3 (Cont.) Release 9.0 and Release 9.2 Optimizer Features

Features	Release		
	9.0.0	9.0.1	9.2.0
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

Parameter type Integer

Default value 0

Parameter class Dynamic: ALTER SESSION

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:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter

OPTIMIZER_INDEX_COST_ADJ

Parameter type	Integer
Default value	100
Parameter class	Dynamic: ALTER SESSION
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: *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter, and on its relationship to "OPTIMIZER_INDEX_CACHING"

OPTIMIZER_MAX_PERMUTATIONS

Parameter type	Integer
Default value	If OPTIMIZER_FEATURES_ENABLE is set to 9.0.0 or higher, then 2000 If OPTIMIZER_FEATURES_ENABLE is set to 8.1.7 or lower, then 80000
Parameter class	Dynamic: ALTER SESSION
Range of values	4 to 80000

OPTIMIZER_MAX_PERMUTATIONS restricts the number of permutations of the tables the optimizer will consider in queries with joins. Such a restriction ensures

that the parse time for the query stays within acceptable limits. However, a slight risk exists that the optimizer will overlook a good plan it would otherwise have found.

The default value for this parameter is 80000, which corresponds to no limitation at all. Setting this parameter to a value less than 1000 normally ensures parse times of a few seconds or less.

OPTIMIZER_MODE

Parameter type	String
Syntax	`OPTIMIZER_MODE =` `{first_rows_[1 | 10 | 100 | 1000] | first_rows | all_rows | choose | rule}`
Default value	`choose`
Parameter class	Dynamic: `ALTER SESSION`

`OPTIMIZER_MODE` establishes the default behavior for choosing an optimization approach for the instance.

Values:

- `rule`

  The optimizer chooses a rule-based approach for all SQL statements regardless of the presence of statistics.

- `choose`

  The optimizer chooses between a cost-based approach and a rule-based approach based on whether statistics are available.

  If the data dictionary contains statistics for at least one of the accessed tables, then the optimizer uses a cost-based approach and optimizes with a goal of best throughput. If the data dictionary contains only some statistics, then the cost-based approach is used, and the optimizer must guess the statistics for the subjects without any statistics. This can result in sub-optimal execution plans. If the data dictionary contains no statistics for any of the accessed tables, then the optimizer uses a rule-based approach.

- `first_rows_`*n*

The optimizer uses a cost-based approach, regardless of the presence of statistics, 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 regardless of the presence of statistics and optimizes with a goal of best throughput (minimum resource use to complete the entire statement).

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- *Oracle9i Database Concepts* and *Oracle9i Database Performance Tuning Guide and Reference* for more information about the optimizer

ORACLE_TRACE_COLLECTION_NAME

Parameter type	String
Syntax	`ORACLE_TRACE_COLLECTION_NAME = collection_name`
Default value	There is no default value.
Parameter class	Static
Range of values	Valid collection name up to 16 characters long (except for platforms that enforce 8-character file names)

A **collection** is data collected for events that occurred while an instrumented product was running. `ORACLE_TRACE_COLLECTION_NAME` specifies the Oracle Trace collection name for this instance. Oracle also uses this parameter in the output file names (collection definition file .cdf and data collection file .dat). If you set `ORACLE_TRACE_ENABLE` to `true`, setting this value to a non-null string will start a default Oracle Trace collection that will run until this value is set to null again.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter
- "ORACLE_TRACE_ENABLE" on page 1-117

ORACLE_TRACE_COLLECTION_PATH

Parameter type	String
Syntax	ORACLE_TRACE_COLLECTION_PATH = *pathname*
Default value	Operating system-specific
Parameter class	Static
Range of values	Full directory pathname

ORACLE_TRACE_COLLECTION_PATH specifies the directory pathname where the Oracle Trace collection definition (.cdf) and data collection (.dat) files are located. If you accept the default, the Oracle Trace .cdf and .dat files will be located in *ORACLE_HOME*/otrace/admin/cdf.

You must specify this parameter if you set the ORACLE_TRACE_FACILITY_NAME parameter.

See Also: *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter

ORACLE_TRACE_COLLECTION_SIZE

Parameter type	Integer
Default value	5242880
Parameter class	Static
Range of values	0 to 4294967295

ORACLE_TRACE_COLLECTION_SIZE specifies (in bytes) the maximum size of the Oracle Trace collection file (.dat). Once the collection file reaches this maximum, the collection is disabled. A value of 0 means that the file has no size limit.

See Also: *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter

ORACLE_TRACE_ENABLE

Parameter type	Boolean
Default value	`false`
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`

To enable Oracle Trace collections for the server, set `ORACLE_TRACE_ENABLE` to `true`. This setting alone does not start an Oracle Trace collection, but it allows Oracle Trace to be used for the server.

With `ORACLE_TRACE_ENABLE` set to `true`, you can perform Oracle Trace collection of server event data in any of the following ways:

- By using Oracle Trace Manager, which is supplied with the Oracle Diagnostic Pack
- By using the Oracle Trace command line interface, which is supplied with the server
- By specifying a collection name in the `ORACLE_TRACE_COLLECTION_NAME` parameter

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter
- "ORACLE_TRACE_COLLECTION_NAME" on page 1-115

ORACLE_TRACE_FACILITY_NAME

Parameter type	String

Syntax	`ORACLE_TRACE_FACILITY_NAME = {ORACLED | ORACLEE | ORACLESM | ORACLEC}`
Default value	`ORACLED`
Parameter class	Static

`ORACLE_TRACE_FACILITY_NAME` specifies the event set that Oracle Trace collects. The value of this parameter, followed by the .fdf extension, is the name of the Oracle Trace product definition file. That file must be located in the directory specified by the `ORACLE_TRACE_FACILITY_PATH` parameter. The product definition file contains definition information for all the events and data items that can be collected for products that use the Oracle Trace data collection API.

The Oracle server has multiple event sets and therefore multiple product definition files:

- `ORACLE` is the `ALL` event set
- `ORACLED` is the `DEFAULT` event set
- `ORACLEE` is the `EXPERT` event set
- `ORACLESM` is the `SUMMARY` event set
- `ORACLEC` is the `CACHEIO` event set

Oracle Corporation recommends that you use the `DEFAULT` event set for Oracle server collections, `ORACLED`.

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter
- "ORACLE_TRACE_COLLECTION_PATH" on page 1-116

ORACLE_TRACE_FACILITY_PATH

Parameter type	String
Syntax	`ORACLE_TRACE_FACILITY_PATH = pathname`
Default value	Operating system-specific
Parameter class	Static

Range of values Full directory pathname

ORACLE_TRACE_FACILITY_PATH specifies the directory pathname where Oracle Trace facility definition files are located. On Solaris, the default path is *ORACLE_HOME*/otrace/admin/fdf/. On NT, the default path is %OTRACE80%\ADMIN\FDF\.

> **See Also:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on the Oracle Trace facility and on setting this parameter

OS_AUTHENT_PREFIX

Parameter type	String
Syntax	OS_AUTHENT_PREFIX = *authentication_prefix*
Default value	OPS$
Parameter class	Static

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

Parameter type	Boolean
Default value	`false`
Parameter class	Static
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:

- *Oracle9i Database Administrator's Guide* and *Oracle Advanced Security Administrator's Guide* for more information on roles and on setting this parameter
- "MAX_ENABLED_ROLES" on page 1-94 and "REMOTE_OS_ROLES" on page 1-140

PARALLEL_ADAPTIVE_MULTI_USER

Parameter type	Boolean
Default value	Derived from the value of PARALLEL_AUTOMATIC_TUNING
Parameter class	Dynamic: 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 parameter is set to TRUE

Tables and hints use the default degree of parallelism.

> **See Also:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution and on the algorithm described here

PARALLEL_AUTOMATIC_TUNING

Parameter type Boolean

Default value `false`

Parameter class Static

Range of values `true | false`

> **Note:** This parameter applies to parallel execution in exclusive mode as well as in the Oracle9*i* Real Application Clusters environment.

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, 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 left unset.

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.

> **See Also:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution

PARALLEL_EXECUTION_MESSAGE_SIZE

Parameter type	Integer
Default value	Operating system-dependent
Parameter class	Static
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 2148bytes 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

Parameter type	String

Syntax	PARALLEL_INSTANCE_GROUP = *group_name*
Default value	A group consisting of all instances currently active
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM
Range of values	Any group name specified in the INSTANCE_GROUPS parameter of any active instance
Real Application Clusters	Different instances can have different values.

PARALLEL_INSTANCE_GROUP is an Oracle9*i* 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:** *Oracle9i Real Application Clusters Administration* for more information on parallel query execution in an Oracle9*i* Real Application Clusters environment

PARALLEL_MAX_SERVERS

Parameter type	Integer
Default value	Derived from the values of CPU_COUNT, PARALLEL_AUTOMATIC_TUNING, and PARALLEL_ADAPTIVE_MULTI_USER
Parameter class	Static
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 the Oracle9*i* 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: *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution

PARALLEL_MIN_PERCENT

Parameter type	Integer
Default value	0
Parameter class	Dynamic: ALTER SESSION
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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution
- "PARALLEL_MAX_SERVERS" on page 1-123, "PARALLEL_MIN_SERVERS" on page 1-125, and "PARALLEL_ADAPTIVE_MULTI_USER" on page 1-120

PARALLEL_MIN_SERVERS

Parameter type	Integer
Default value	0
Parameter class	Static
Range of values	0 to value of PARALLEL_MAX_SERVERS
Real Application Clusters	Multiple instances can have different values.

Note: This parameter applies to parallel execution in exclusive mode as well as in the Oracle9*i* 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: *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution

PARALLEL_THREADS_PER_CPU

Parameter type	Integer
Default value	Operating system-dependent, usually 2
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any nonzero number

> **Note:** This parameter applies to parallel execution in exclusive mode as well as in the Oracle9*i* 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:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on tuning parallel execution

PARTITION_VIEW_ENABLED

Parameter type	Boolean
Default value	false
Parameter class	Dynamic: ALTER SESSION
Range of values	true | false

> **Note:** Oracle Corporation recommends that you use partitioned tables (available starting with Oracle8) rather than partition views. Partition views are supported for backward compatibility only.

PARTITION_VIEW_ENABLED specifies whether the optimizer uses partition views. If you set this parameter to true, the optimizer prunes (or skips) unnecessary table accesses in a partition view and alters the way it computes statistics on a partition view from statistics on underlying tables.

> **See Also:** *Oracle9i Database Concepts* for information on the advantages of partitioned tables and how to convert partition views to partitioned tables

PGA_AGGREGATE_TARGET

Parameter type	Big integer
Syntax	PGA_AGGREGATE_TARGET = *integer* [K | M | G]
Default value	0 (automatic memory management is turned OFF by default)
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Minimum: 10 MB Maximum: 4096 GB - 1

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_COMPILER_FLAGS

Parameter type	String
Syntax	`PLSQL_COMPILER_FLAGS = { [DEBUG | NON_DEBUG] [INTERPRETED | NORMAL] }`
Default value	`INTERPRETED, NON_DEBUG`
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM`

`PLSQL_COMPILER_FLAGS` is a parameter used by the PL/SQL compiler. It specifies a list of compiler flags as a comma-separated list of strings.

Values:

- `INTERPRETED`

  PL/SQL modules will be compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine.

- `NATIVE`

  PL/SQL modules (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 modules will be compiled with PROBE debug symbols.

The following combinations of flags are invalid specifications for the `PLSQL_COMPILER_FLAGS` parameter:

- `NATIVE` & `DEBUG`
- `NATIVE` & `INTERPRETED`

When the value of the parameter is changed, it has no effect on PL/SQL modules 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_NATIVE_C_COMPILER

Parameter type	String
Syntax	PLSQL_NATIVE_C_COMPILER = *pathname*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any valid path name

PLSQL_NATIVE_C_COMPILER specifies the full path name of a C compiler which is used to compile the generated C file into an object file.

PLSQL_NATIVE_C_COMPILER is optional. The platform-specific make file that is shipped for each platform contains a default value for this parameter. If a value is specified for this parameter, it will override the default value in the make file.

PLSQL_NATIVE_LIBRARY_DIR

Parameter type	String
Syntax	PLSQL_NATIVE_LIBRARY_DIR = *directory*
Default value	There is no default value.
Parameter class	Dynamic: 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

Parameter type	Integer
Default value	0
Parameter class	Dynamic: 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_NATIVE_LINKER

Parameter type	String
Syntax	PLSQL_NATIVE_LINKER = *pathname*
Default value	There is no default value.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any valid path name

PLSQL_NATIVE_LINKER specifies the full path name of a linker such as ld in UNIX or GNU ld which is used to link the object file into a shared object or DLL.

PLSQL_NATIVE_LINKER is optional. The platform-specific make file that is shipped for each platform contains a default value for this parameter. If a value is specified for this parameter, it will override the default value in the make file.

PLSQL_NATIVE_MAKE_FILE_NAME

Parameter type	String
Syntax	PLSQL_NATIVE_MAKE_FILE_NAME = *pathname*
Default value	There is no default value.

Parameter class Dynamic: ALTER SYSTEM

Range of values Any valid path name

PLSQL_NATIVE_MAKE_FILE_NAME specifies the full path name of a make file. The make utility (specified by *PLSQL_NATIVE_MAKE_UTILITY*) uses this make file to generate the shared object or DLL.

A port-specific make file is shipped for each platform that contains the rules for the make utility to generate DLLs on that platform.

PLSQL_NATIVE_MAKE_UTILITY

Parameter type String

Syntax PLSQL_NATIVE_MAKE_UTILITY = *pathname*

Default value There is no default value.

Parameter class Dynamic: ALTER SYSTEM

Range of values Any valid path name

PLSQL_NATIVE_MAKE_UTILITY specifies the full path name of a make utility such as make in UNIX or gmake (GNU make). The make utility is needed to generate the shared object or DLL from the generated C source.

PLSQL_V2_COMPATIBILITY

Parameter type Boolean

Default value false

Parameter class Dynamic: 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

PRE_PAGE_SGA

Parameter type	Boolean
Default value	`false`
Parameter class	Static
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

Parameter type	Integer
Default value	Derived from PARALLEL_MAX_SERVERS
Parameter class	Static
Range of values	6 to operating system-dependent
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:

- *Oracle9i Database Administrator's Guide* for information on setting this parameter in exclusive mode
- *Oracle9i Real Application Clusters Administration* for information on setting this parameter in an Oracle9*i* Real Application Clusters environment
- Your operating system-specific Oracle documentation for the range of values

QUERY_REWRITE_ENABLED

Parameter type	String
Syntax	`QUERY_REWRITE_ENABLED = {force | true | false}`
Default value	`false`
Parameter class	Dynamic: 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:

- `true`

  Oracle costs the query with rewrite and without rewrite and chooses the method with the lower cost.

- `false`

  Oracle does not use rewrite.

- `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:

- *Oracle9i Data Warehousing Guide* for information on query rewrite of materialized views
- *Oracle9i Database Performance Tuning Guide and Reference* and "OPTIMIZER_MODE" on page 1-114 for information on cost-based optimization

QUERY_REWRITE_INTEGRITY

Parameter type	String
Syntax	`QUERY_REWRITE_INTEGRITY = {stale_tolerated | trusted | enforced}`
Default value	`enforced`
Parameter class	Dynamic: `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

Parameter type	X.500 Distinguished Name
Default value	There is no default value.
Parameter class	Static
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

Parameter type	Boolean
Default value	`false`
Parameter class	Static
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:** *Oracle9i Database Administrator's Guide* for information on the consequences of delaying access of datafiles in read-only tablespaces

RECOVERY_PARALLELISM

Parameter type	Integer
Default value	Operating system-dependent
Parameter class	Static
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:

- *Oracle9i Database Performance Tuning Guide and Reference* for information on setting this parameter in exclusive mode
- *Oracle9i Real Application Clusters Administration* for information on setting this parameter in an Oracle9*i* Real Application Clusters environment

REMOTE_ARCHIVE_ENABLE

Parameter type	String
Syntax	REMOTE_ARCHIVE_ENABLE = {receive [, send] | false | true}
Default value	true
Parameter class	Static

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

Parameter type	String
Syntax	`REMOTE_DEPENDENCIES_MODE = {TIMESTAMP | SIGNATURE}`
Default value	`TIMESTAMP`
Parameter class	Dynamic: `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: *Oracle9i Application Developer's Guide - Fundamentals* for information about the consequences of the settings of this parameter

REMOTE_LISTENER

Parameter type	String
Syntax	`REMOTE_LISTENER = network_name`
Default value	There is no default value.
Parameter class	Dynamic: `ALTER SYSTEM`

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:

- *Oracle9i Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle9i 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

Parameter type	String
Syntax	REMOTE_LOGIN_PASSWORDFILE= {NONE | SHARED | EXCLUSIVE}
Default value	NONE
Parameter class	Static
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 Oracle9*i* Real Application Clusters

---

See Also: *Oracle9i Database Administrator's Guide*for more information about secure connections for privileged users

REMOTE_OS_AUTHENT

Parameter type	Boolean
Default value	`false`
Parameter class	Static
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 1-119

REMOTE_OS_ROLES

Parameter type	Boolean
Default value	`false`
Parameter class	Static
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:

- *Oracle9i Database Administrator's Guide* for more information on setting this parameter
- "OS_ROLES" on page 1-120

REPLICATION_DEPENDENCY_TRACKING

Parameter type	Boolean
Default value	`true`
Parameter class	Static
Range of values	`true | false`

`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: *Oracle9i Replication* for more information on parallel propagation dependency tracking

RESOURCE_LIMIT

Parameter type	Boolean

Default value	`false`
Parameter class	Dynamic: `ALTER SYSTEM`
Range of values	`true | false`

`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: *Oracle9i Database Administrator's Guide* and *Oracle9i SQL Reference* for more information on setting resource limits for profiles

RESOURCE_MANAGER_PLAN

Parameter type	String
Syntax	`RESOURCE_MANAGER_PLAN = plan_name`
Default value	There is no default value.
Parameter class	Dynamic: `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:

- *Oracle9i Database Administrator's Guide* for information on resource plans
- *Oracle9i Supplied PL/SQL Packages and Types Reference* for information on the DBMS_RESOURCE_MANAGER and DBMS_RESOURCE_MANAGER_PRIVS packages
- "DBA_RSRC_PLANS" on page 2-201, "DBA_RSRC_PLAN_DIRECTIVES" on page 2-200, and the various V$RSRC_ dynamic performance views in Chapter 3 for information on existing resource plans

ROLLBACK_SEGMENTS

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
Parameter class	Static
Range of values	Any rollback segment names listed in DBA_ROLLBACK_SEGS except SYSTEM
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 TRANSACTIONS / 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:

- *Oracle9i Database Administrator's Guide* for more information on setting this parameter
- *Oracle9i Real Application Clusters Deployment and Performance* for information on setting this parameter in an Oracle9*i* Real Application Clusters environment
- "DBA_ROLLBACK_SEGS" on page 2-198

ROW_LOCKING

Parameter type	String
Syntax	ROW_LOCKING = {ALWAYS | DEFAULT | INTENT}
Default value	ALWAYS
Parameter class	Static
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

ROW_LOCKING specifies whether row locks are acquired during UPDATE operations.

Values:

- ALWAYS and DEFAULT

  Row locks are acquired. Table locks are not acquired.

- INTENT

  In addition to row locks, restricted table-level locks are acquired during UPDATE operations.

The acquisition of table-level locks is a way to ensure data concurrency. However, Oracle Corporation recommends that you use the SQL statement LOCK TABLE for that purpose rather than changing the setting of this parameter from its default setting of ALWAYS.

SERIAL _REUSE

Parameter type	String
Syntax	SERIAL_REUSE = {DISABLE | SELECT | DML | PLSQL | ALL}
Default value	DISABLE
Parameter class	Static

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 1-31

SERVICE_NAMES

Parameter type	String
Syntax	SERVICE_NAMES = *db_service_name* [, *db_service_name* [,...]]
Default value	DB_NAME.DB_DOMAIN if defined
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Any ASCII string or comma-separated list of string names

SERVICE_NAMES specifies one or more names for the database service to which this instance connects. 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.

In an Oracle9*i* Real Application Clusters environment, you must set this parameter for every instance.

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, Oracle uses the domain of your local database as it currently exists in the data dictionary.

See Also:

- *Oracle9i Net Services Administrator's Guide* for more information on this parameter and its settings
- "DB_DOMAIN" on page 1-39

SESSION_CACHED_CURSORS

Parameter type	Integer
Default value	0

Parameter class	Dynamic: `ALTER SESSION`
Range of values	0 to operating system-dependent
Real Application Clusters	Multiple instances can have different values.

`SESSION_CACHED_CURSORS` lets you specify 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.

> **See Also:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter and its uses

SESSION_MAX_OPEN_FILES

Parameter type	Integer
Default value	`10`
Parameter class	Static
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:**
>
> - *Oracle9i Application Developer's Guide - Large Objects (LOBs)* for information on LOBs in general and BFILEs in particular
> - *Oracle9i Supplied 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

Parameter type	Integer
Default value	Derived: (1.1 * PROCESSES) + 5
Parameter class	Static
Range of values	1 to 2^{31}

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 Corporation recommends that you adjust the value of SESSIONS to approximately 1.1 * total number of *connections*.

See Also:

- *Oracle9i Database Concepts* for more information on memory structures and processes
- "ENQUEUE_RESOURCES" on page 1-55 and "TRANSACTIONS" on page 1-164

SGA_MAX_SIZE

Parameter type	Big integer
Syntax	SGA_MAX_SIZE = *integer* [K | M | G]

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.
Parameter class	Static
Range of values	0 to operating system-dependent

SGA_MAX_SIZE specifies the maximum size of SGA for the lifetime of the instance.

SHADOW_CORE_DUMP

Parameter type	String
Syntax	SHADOW_CORE_DUMP = {partial | full | none}
Default value	partial
Parameter class	Static

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 1-18

SHARED_MEMORY_ADDRESS

Parameter type	Integer

Default value	0
Parameter class	Static

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 1-64

SHARED_POOL_RESERVED_SIZE

Parameter type	Big integer
Syntax	SHARED_POOL_RESERVED_SIZE = *integer* [K | M | G]
Default value	5% of the value of SHARED_POOL_SIZE
Parameter class	Static
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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- "SHARED_POOL_SIZE" on page 1-151

SHARED_POOL_SIZE

Parameter type	Big integer
Syntax	SHARED_POOL_SIZE = *integer* [K | M | G]
Default value	32-bit platforms: 8 MB, rounded up to the nearest granule size 64-bit platforms: 64 MB, rounded up to the nearest granule size
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- *Oracle9i Database Migration* for information on parallel execution message buffers
- "PARALLEL_AUTOMATIC_TUNING" on page 1-121 and "V$SGASTAT" on page 3-134

SHARED_SERVER_SESSIONS

Parameter type	Integer
Default value	Derived: the lesser of CIRCUITS and SESSIONS - 5
Parameter class	Static
Range of values	0 to SESSIONS - 5

SHARED_SERVER_SESSIONS specifies the total number of shared server architecture user sessions to allow. Setting this parameter enables you to reserve user sessions for dedicated servers.

> **See Also:** *Oracle9i Database Concepts* for more information on memory structures and processes

SHARED_SERVERS

Parameter type	Integer
Default value	If you are using shared server architecture, then the value is 1. If you are not using shared server architecture, then the value is 0.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Operating system-dependent

SHARED_SERVERS specifies the number of server processes that you want to create when an instance is started up. If system load decreases, this minimum number of servers is maintained. Therefore, you should take care not to set SHARED_SERVERS too high at system startup.

> **See Also:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter

SORT_AREA_RETAINED_SIZE

Parameter type	Integer
Default value	Derived from SORT_AREA_SIZE
Parameter class	Dynamic: 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- "SORT_AREA_SIZE" on page 1-153

SORT_AREA_SIZE

Parameter type	Integer
Default value	65536

Parameter class	Dynamic: 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter
- *Oracle9i 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 1-152

SPFILE

Parameter type	String
Syntax	SPFILE = *spfile_name*
Default value	*ORACLE_HOME*/dbs/spfile.ora
Parameter class	Static
Range of values	Any valid SPFILE
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 a PFILE.

See Also: *Oracle9i Database Performance Tuning Guide and Reference* for more information about creating the SPFILE

SQL_TRACE

Parameter type	Boolean

Default value	`false`
Parameter class	Static
Range of values	`true | false`

The value of SQL_TRACE disables or enables 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, 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

Note: If you change the value of this parameter by using the ALTER SESSION SET SQL_TRACE statement, the resulting change will not be reflected in the V$PARAMETER dynamic performance view. Therefore, this parameter is not considered dynamic. For information on the ALTER SESSION statement, see *Oracle9i SQL Reference*.

See Also: *Oracle9i Database Performance Tuning Guide and Reference*for more information about performance diagnostic tools

SQL92_SECURITY

Parameter type	Boolean
Default value	`false`

Parameter class	Static
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` lets you specify whether users must have been granted the `SELECT` object privilege in order to execute such `UPDATE` or `DELETE` statements.

STANDBY_ARCHIVE_DEST

Parameter type	String
Syntax	`STANDBY_ARCHIVE_DEST = filespec`
Default value	Operating system-specific
Parameter class	Dynamic: `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:

- *Oracle9i Data Guard Concepts and Administration* for more information on setting this parameter and on managed recovery mode in general
- "LOG_ARCHIVE_DEST" on page 1-75 and "V$ARCHIVE_DEST" on page 3-5

STANDBY_FILE_MANAGEMENT

Parameter type	String
Syntax	STANDBY_FILE_MANAGEMENT = {MANUAL | AUTO}
Default value	MANUAL
Parameter class	Dynamic: 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:** *Oracle9i Data Guard Concepts and Administration* for more information about setting this parameter

STAR_TRANSFORMATION_ENABLED

Parameter type	String
Syntax	STAR_TRANSFORMATION_ENABLED = {TEMP_DISABLE | TRUE | FALSE}
Default value	FALSE

Parameter class Dynamic: ALTER SESSION

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:

- *Oracle9i Database Concepts* for information on star queries
- *Oracle9i Database Performance Tuning Guide and Reference* for information on enabling star query

STATISTICS_LEVEL

Parameter type	String
Syntax	STATISTICS_LEVEL = {ALL | TYPICAL | BASIC}
Default value	TYPICAL
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM

STATISTICS_LEVEL sets the statistics collection level of the database.

TAPE_ASYNCH_IO

Parameter type Boolean

Default value	`true`
Parameter class	Static
Range of values	`true | false`

`TAPE_ASYNCH_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 Corporation recommends that you leave this parameter set to its default. However, if the asynchronous I/O implementation is not stable, you can set `TAPE_ASYNCH_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:** *Oracle9i Database Performance Tuning Guide and Reference* for more information on asynchronous I/O and on setting this parameter

THREAD

Parameter type	Integer
Default value	`0`
Parameter class	Static
Range of values	0 to the maximum number of enabled threads
Real Application Clusters	If specified, multiple instances must have different values.

`THREAD` is an Oracle9*i* 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:** *Oracle9i Real Application Clusters Administration* and *Oracle9i SQL Reference*

TIMED_OS_STATISTICS

Parameter type	Integer
Default value	If STATISTICS_LEVEL is set to ALL, then 5
	If STATISTICS_LEVEL is set to BASIC or TYPICAL, then 0
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM
Range of values	Unlimited

TIMED_OS_STATISTICS specifies the interval (in seconds) 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 Corporation 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

Parameter type	Boolean
Default value	If STATISTICS_LEVEL is set to TYPICAL or ALL, then true If STATISTICS_LEVEL is set to BASIC, then false
Parameter class	Dynamic: ALTER SESSION, ALTER SYSTEM
Range of values	true | false

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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on setting this parameter and on performance diagnostic tools in general
- Appendix C, "Statistics Descriptions" indicates which statistics depend on the setting of this parameter.

TRACE_ENABLED

Parameter type	Boolean
Default value	true

Parameter class	Dynamic: ALTER SYSTEM
Range of values	true | false
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 *Oracle9i Real Application Clusters Deployment and Performance* 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

Parameter type	String
Syntax	TRACEFILE_IDENTIFIER = "*traceid*"
Default value	There is no default value.
Parameter class	Dynamic: 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.

TRANSACTION_AUDITING

Parameter type	Boolean
Default value	true
Parameter class	Dynamic: ALTER SYSTEM ... DEFERRED
Range of values	true | false

If TRANSACTION_AUDITING is true, Oracle generates a special redo record that contains the user logon name, username, the session ID, some operating system information, and client information. For each successive transaction, Oracle generates a record that contains only the session ID. These subsequent records link back to the first record, which also contains the session ID.

These records might be useful if you are using a redo log analysis tool. You can access the records by dumping the redo log.

If TRANSACTION_AUDITING is false, no redo record will be generated.

TRANSACTIONS

Parameter type	Integer

Default value	Derived: (1.1 * SESSIONS)
Parameter class	Static
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:

- *Oracle9i Database Administrator's Guide* for information on setting this parameter
- *Oracle9i Real Application Clusters Administration* for the relationship of this parameter to the number of rollback segments

TRANSACTIONS_PER_ROLLBACK_SEGMENT

Parameter type	Integer
Default value	5
Parameter class	Static
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:

- *Oracle9i Database Administrator's Guide* for information on setting this parameter.
- *Oracle9i Real Application Clusters Administration* for how Oracle acquires rollback segments.
- Your operating system-specific Oracle documentation for the range of values for this parameter.

UNDO_MANAGEMENT

Parameter type	String
Syntax	UNDO_MANAGEMENT = {MANUAL | AUTO}
Default value	MANUAL
Parameter class	Static
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

Parameter type	Integer
Default value	900
Parameter class	Dynamic: 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 amount of committed undo information to retain in the database. You can use UNDO_RETENTION to satisfy queries that require old undo information to rollback changes to produce older images of data blocks. You can set the value at instance startup.

The UNDO_RETENTION parameter works best if the current undo tablespace has enough space for the active transactions. If an active transaction needs undo space and the undo tablespace does not have any free space, then the system will start reusing undo space that would have been retained. This may cause long queries to fail. Be sure to allocate enough space in the undo tablespace to satisfy the space requirement for the current setting of this parameter.

> **See Also:** *Oracle9i SQL Reference* for more information about creating undo tablespaces

UNDO_SUPPRESS_ERRORS

Parameter type	Boolean
Default value	`false`
Parameter class	Dynamic: `ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`

UNDO_SUPPRESS_ERRORS enables users to suppress errors while executing manual undo management mode operations (for example, `ALTER ROLLBACK SEGMENT ONLINE`) in automatic undo management mode. Setting this parameter enables users to use the undo tablespace feature before all application programs and scripts are converted to automatic undo management mode. For example, if you have a tool that uses `SET TRANSACTION USE ROLLBACK SEGMENT` statement, you can add the statement "`ALTER SESSION SET UNDO_SUPPRESS_ERRORS = true`" to the tool to suppress the ORA-30019 error.

If you want to run in automatic undo management mode, ensure that your tools or applications are updated to run in automatic undo management mode.

UNDO_TABLESPACE

Parameter type	String

Syntax	UNDO_TABLESPACE = *undoname*
Default value	The first available undo tablespace in the database.
Parameter class	Dynamic: ALTER SYSTEM
Range of values	Legal name of an existing undo tablespace
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, 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: *Oracle9i SQL Reference*.

USE_INDIRECT_DATA_BUFFERS

Parameter type	Boolean
Default value	false
Parameter class	Static
Range of values	true | false

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 1-74, "SHARED_MEMORY_ADDRESS" on page 1-149, and "HI_SHARED_MEMORY_ADDRESS" on page 1-64, which are other parameters that control how the SGA uses memory.
- *Oracle9i Database Concepts* for more information about the SGA.

USER_DUMP_DEST

Parameter type	String
Syntax	USER_DUMP_DEST = {*pathname* | *directory*}
Default value	Operating system-dependent
Parameter class	Dynamic: 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:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information about the use of trace files
- Your operating system-specific Oracle documentation for the range of values

UTL_FILE_DIR

Parameter type	String
Syntax	UTL_FILE_DIR = *pathname*

Default value	There is no default value.
Parameter class	Static
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

Parameter type	String
Syntax	WORKAREA_SIZE_POLICY = {AUTO | MANUAL}
Default value	If PGA_AGGREGATE_TARGET is set, then AUTO
	If PGA_AGGREGATE_TARGET is not set, then MANUAL
Parameter class	Dynamic: 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.

2

Static Data Dictionary Views

This chapter 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 a user is granted new privileges). Oracle also maintains tables that monitor ongoing database activity. These **dynamic performance tables** are described in Chapter 3, "Dynamic Performance (V$) 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.

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_REPAUDIT_ATTRIBUTE	DBA_REPAUDIT_ATTRIBUTE	USER_REPAUDIT_ATTRIBUTE
ALL_REPAUDIT_COLUMN	DBA_REPAUDIT_COLUMN	USER_REPAUDIT_COLUMN
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_PARMS	DBA_REPCAT_TEMPLATE_PARMS	USER_REPCAT_TEMPLATE_PARMS
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

Table 2–1 (Cont.) Oracle Replication Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
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
ALL_REPSCHEMA	DBA_REPSCHEMA	USER_REPSCHEMA
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: *Oracle9i Replication Management API Reference* for information about these 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

Table 2–2 (Cont.) Oracle Workspace Manager Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
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:** *Oracle9i Application Developer's Guide - Workspace Manager* for information about these 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:** *Oracle9i Recovery Manager Reference* for information about these views

The remainder of this chapter lists the data dictionary views in alphabetical order.

ALL_ALL_TABLES

ALL_ALL_TABLES describes all 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 all 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
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 entry belongs
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER		Minimum percentage of used space in a block
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_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
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
BACKED_UP	VARCHAR2(1)		Whether the table has been backed up since the 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

Column	Datatype	NULL	Description
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)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or 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 this table is partitioned (YES | NO)
IOT_TYPE	VARCHAR2(12)		If an index organized table, whether an overflow segment has been specified (OVERFLOW) or not (NULL)
OBJECT_ID_TYPE	VARCHAR2(16)		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)		Whether the table is temporary
SECONDARY	VARCHAR2(1)		Whether the table is a secondary object created by the ODCIIndexCreate method of the Oracle9*i* Data Cartridge to contain the contents of a domain index (Y | N)
NESTED	VARCHAR2(3)		Whether this is a nested table
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the table
ROW_MOVEMENT	VARCHAR2(8)		If a partitioned table, whether row movement is enabled or disabled

Column	Datatype	NULL	Description
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)		Whether statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: 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 Oracle 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

ALL_APPLY

ALL_APPLY displays information about the apply processes that dequeue 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 the apply process dequeues from
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue the apply process dequeues from
APPLY_CAPTURED	VARCHAR2(3)		Indicates whether applying captured messages (YES) or enqueued messages (NO)
RULE_SET_NAME	VARCHAR2(30)		Rule set used by apply process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set
APPLY_USER	VARCHAR2(30)	NOT NULL	Current user who is applying the messages
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes will be applied
APPLY_TAG	RAW(2000)		Tag associated with DDL and DML change records that will be applied

Column	Datatype	NULL	Description
DDL_HANDLER	VARCHAR2(92)		Name of the user-specified ddl handler
MESSAGE_HANDLER	VARCHAR2(92)		User-specified procedure to handle messages other than DDL and DML messages
STATUS	VARCHAR2(8)		Status of the apply process: ■ DISABLED ■ ENABLED ■ ABORTED

See Also: "DBA_APPLY" on page 2-146

ALL_APPLY_CONFLICT_COLUMNS

ALL_APPLY_CONFLICT_COLUMNS displays information about conflict resolution on the tables accessible to the current user.

Related View

- DBA_APPLY_CONFLICT_COLUMNS displays information about conflict resolution on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
METHOD_NAME	VARCHAR2(92)		Name of the method used to resolve conflict
RESOLUTION_COLUMN	VARCHAR2(4000)		Name of the column used to resolve conflict
COLUMN_NAME	VARCHAR2(30)		Name of the column that is to be considered as part of a group to resolve conflict
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes will be applied

See Also: "DBA_APPLY_CONFLICT_COLUMNS" on page 2-146

ALL_APPLY_DML_HANDLERS

ALL_APPLY_DML_HANDLERS displays information about the DML handler on the tables accessible to the current user.

Related View

- DBA_APPLY_DML_HANDLERS displays information about the DML handler on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OPERATION_NAME	VARCHAR2(12)		Name of the DML operation
USER_PROCEDURE	VARCHAR2(92)		Name of the DML handler specified by the user
ERROR_HANDLER	VARCHAR2(1)		Indicates whether the user procedure handles only DML errors (Y) or the entire DML operation (N)
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes will be applied

See Also: "DBA_APPLY_DML_HANDLERS" on page 2-147

ALL_APPLY_ERROR

ALL_APPLY_ERROR displays information about error transactions that were generated after dequeuing from queues accessible to the current user.

Related View

- DBA_APPLY_ERROR displays information about error transactions.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)		Name of the apply process at the local site which processed the transaction
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue at the local site from which the transaction was dequeued
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue at the local site from which the transaction was dequeued
LOCAL_TRANSACTION_ID	VARCHAR2(22)		Local transaction ID for the error creation 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 SCN for the transaction at the source database
MESSAGE_NUMBER	NUMBER		Identifier for the message in the transaction that raised an error

Column	Datatype	NULL	Description
ERROR_NUMBER	NUMBER		Error number
ERROR_MESSAGE	VARCHAR2(4000)		Error message
RECIPIENT_ID	NUMBER		User ID of the original recipient
RECIPIENT_NAME	VARCHAR2(30)		Name of the original recipient
MESSAGE_COUNT	NUMBER		Total number of messages inside the error transaction

See Also: "DBA_APPLY_ERROR" on page 2-147

ALL_APPLY_KEY_COLUMNS

ALL_APPLY_KEY_COLUMNS displays information about alternative key columns for STREAMS tables accessible to the current user.

Related View

- DBA_APPLY_KEY_COLUMNS displays information about alternative key columns for all STREAMS tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name of the object
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes will be applied

See Also: "DBA_APPLY_KEY_COLUMNS" on page 2-147

ALL_APPLY_PARAMETERS

ALL_APPLY_PARAMETERS displays information about all parameters for each apply process which dequeues from queues accessible to the current user.

Related View

- DBA_APPLY_PARAMETERS displays information about all parameters for the apply process.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply 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 to the default (NO)

See Also: "DBA_APPLY_PARAMETERS" on page 2-148

ALL_APPLY_PROGRESS

ALL_APPLY_PROGRESS displays progress information made by the apply processes that dequeue from queues accessible to the current user.

Related View

DBA_APPLY_PROGRESS displays progress information made by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
APPLIED_MESSAGE_NUMBER	NUMBER	NOT NULL	Message number up to which all transactions have been successfully applied
OLDEST_MESSAGE_NUMBER	NUMBER	NOT NULL	Earliest message number of the transactions currently being applied
APPLY_TIME	DATE		Time at which the message was applied
APPLIED_MESSAGE_CREATE_TIME	DATE		Time at which the message to be applied was created

See Also: "DBA_APPLY_PROGRESS" on page 2-148

ALL_ARGUMENTS

ALL_ARGUMENTS lists all arguments of procedures and functions that are accessible to the user.

Related View

- USER_ARGUMENTS lists all arguments of 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	Name of the owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the procedure or function
PACKAGE_NAME	VARCHAR2(30)		Package name
OBJECT_ID	NUMBER	NOT NULL	Name of the package containing the procedure or function
OVERLOAD	VARCHAR2(40)		Overload unique identifier
ARGUMENT_NAME	VARCHAR2(30)		Name of the argument
POSITION	NUMBER	NOT NULL	Position in argument list, or NULL for function return value
SEQUENCE	NUMBER	NOT NULL	Argument sequence, including all nesting levels
DATA_LEVEL	NUMBER	NOT NULL	Nesting depth of argument for composite types
DATA_TYPE	VARCHAR2(14)		Datatype of the argument
DEFAULT_VALUE	LONG		Default value for the argument
DEFAULT_LENGTH	NUMBER		Length of default value for the argument
IN_OUT	VARCHAR2(9)		Argument direction (IN, OUT, or 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 right of 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)		Username of the owner of the type of the argument
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), 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.

Column	Datatype	NULL	Description
TYPE_LINK	VARCHAR2(128)		Relevant only for package local types when the package identified in the TYPE_NAME column is a remote package. This columns 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.

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 Oracle8*i* 8.1; 0 for types implementing Oracle9*i* 9.0.0.

ALL_BASE_TABLE_MVIEWS

ALL_BASE_TABLE_MVIEWS describes all materialized views using materialized view logs accessible to the current user. A materialized view log can be created for a master or base table or a 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 all materialized views using materialized view logs owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema in which the master table or the master materialized view was created
MASTER	VARCHAR2(30)		Name of the master table or the master materialized view
MVIEW_LAST_REFRESH_TIME	DATE		Date when the materialized view based on the master was last refreshed
MVIEW_ID	NUMBER		Unique identifier of the materialized view that is based on the master

See Also:

- "DBA_BASE_TABLE_MVIEWS" on page 2-156
- "USER_BASE_TABLE_MVIEWS" on page 2-244

ALL_CAPTURE

ALL_CAPTURE displays information about each capture process that stores the captured changes in 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 holding captured changes
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue used for holding captured changes
RULE_SET_NAME	VARCHAR2(30)		Name of the rule set used by the capture process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set
START_SCN	NUMBER		System change number (SCN) from which capturing will be resumed
STATUS	VARCHAR2(8)		Status of the capture process (DISABLED, ENABLED, or ABORTED)

See Also: "DBA_CAPTURE" on page 2-156

ALL_CAPTURE_PARAMETERS

ALL_CAPTURE_PARAMETERS displays information about all parameters for each capture process that stores the captured changes in queues accessible to the current user.

Related View

- DBA_CAPTURE_PARAMETERS displays information about all parameters for the capture process.

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)		(YES/NO) Indicates whether the parameter value was set by the user. If NO, then the parameter is set to the default value.

See Also: "DBA_CAPTURE_PARAMETERS" on page 2-156

ALL_CAPTURE_PREPARED_DATABASE

ALL_CAPTURE_PREPARED_DATABASE displays information about whether the local database is prepared for instantiation.

Related View

- DBA_CAPTURE_PREPARED_DATABASE displays information about whether the local database is prepared for instantiation.

Column	Datatype	NULL	Description
TIMESTAMP	DATE		Time at which the database was ready to be instantiated

See Also: "DBA_CAPTURE_PREPARED_DATABASE" on page 2-157

ALL_CAPTURE_PREPARED_SCHEMAS

ALL_CAPTURE_PREPARED_SCHEMAS displays information about all user schemas at the local database that are prepared for instantiation.

Related View

- DBA_CAPTURE_PREPARED_SCHEMAS displays information about all schemas at the local database that are prepared for instantiation.

Column	Datatype	NULL	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Name of schema prepared for instantiation
TIMESTAMP	DATE		Time at which the schema was ready to be instantiated

See Also: "DBA_CAPTURE_PREPARED_SCHEMAS" on page 2-157

ALL_CAPTURE_PREPARED_TABLES

ALL_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation that are accessible to the current user.

Related View

- DBA_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the table prepared for instantiation
TABLE_NAME	VARCHAR2(30)		Name of the table prepared for instantiation
SCN	NUMBER	NOT NULL	SCN from which changes can be captured
TIMESTAMP	DATE		Time at which the table was ready to be instantiated

See Also: "DBA_CAPTURE_PREPARED_TABLES" on page 2-157

ALL_CATALOG

ALL_CATALOG lists all indexes, tables, clusters, views, synonyms, and sequences accessible to the 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 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
PCT_USED	NUMBER	NOT NULL	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	NOT NULL	Percentage increase in extent size

Column	Datatype	NULL	Description
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)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the cluster
SINGLE_TABLE	VARCHAR2(5)		Whether this is a single-table cluster (Y | N)

ALL_COL_COMMENTS

ALL_COL_COMMENTS lists comments on columns of tables and views accessible to the current user.

Related Views

- DBA_COL_COMMENTS lists comments on all tables and views in the database.
- USER_COL_COMMENTS lists comments on 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

ALL_COL_PRIVS

ALL_COL_PRIVS describes column object grants for which the current user or PUBLIC is the object owner, grantor, or grantee. This view displays OWNER but not TABLE_SCHEMA.

Related Views

- DBA_COL_PRIVS describes all column object grants in the database. This view displays OWNER but not TABLE_SCHEMA.
- USER_COL_PRIVS describes column object grants for which the current user is the object owner, grantor, or grantee. This view displays TABLE_SCHEMA but not OWNER.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object. Displayed only in the DBA_ and USER_ views.
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. Displayed only in the ALL_ view.
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)		Whether privileges were granted with ADMIN OPTION (YES | NO)

ALL_COL_PRIVS_MADE

ALL_COL_PRIVS_MADE lists column object grants for which the current user is object owner or grantor.

Related View

- USER_COL_PRIVS_MADE describes column object grants for which the current user is the grantor. 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

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the 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)		Whether the privilege was granted with ADMIN OPTION (YES | NO)

ALL_COL_PRIVS_RECD

ALL_COL_PRIVS_RECD describes column object grants for which the current user or PUBLIC is the grantee.

Related View

- USER_COL_PRIVS_RECD describes column object grants for which the current user is the grantee. 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 NUL	Name of the owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_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 column
GRANTABLE	VARCHAR2(3)		Whether the privilege was granted with ADMIN OPTION (YES | NO)

ALL_COLL_TYPES

ALL_COLL_TYPES describes all named collection types (varrays and nested tables) accessible to the 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)		Whether the collection
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.
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

Column	Datatype	NULL	Description
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"."atttribute"
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
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
CONS_TYPE_OWNER	VARCHAR2(30)		Owner of the type that the column (or element) is constrained to
CONS_TYPE_NAME	VARCHAR2(30)		Name of the type that the column (or element) is constrained to

Column	Datatype	NULL	Description
CONS_TYPE_ONLY	VARCHAR2(15)		Indicates whether the column (or element) is constrained to ONLY type (Y) or not (N)

See Also:

- "DBA_CONS_OBJ_COLUMNS" on page 2-159
- "USER_CONS_OBJ_COLUMNS" on page 2-245

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
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name of the constraint definition
CONSTRAINT_TYPE	VARCHAR2(1)		Type of constraint definition: ■ 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)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name associated with the table (or view) with constraint definition
SEARCH_CONDITION	LONG		Text of search condition for a check constraint
R_OWNER	VARCHAR2(30)		Owner of table referred to in a referential constraint
R_CONSTRAINT_NAME	VARCHAR2(30)		Name of the unique constraint definition for referenced table
DELETE_RULE	VARCHAR2(9)		Delete rule for a referential constraint (CASCADE or NO ACTION)

Column	Datatype	NULL	Description
STATUS	VARCHAR2(8)		Enforcement status of constraint (ENABLED or DISABLED)
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 *Oracle9i SQL Reference* and *Oracle9i Application Developer's Guide - Fundamentals*
RELY	VARCHAR2(4)		Whether an enabled constraint is enforced or unenforced. **See Also:** the *constraints* in *Oracle9i SQL Reference*
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

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.

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

Column	Datatype	NULL	Description
PACKAGE	VARCHAR2(30)	NOT NULL	Package name of the designated package that can set attributes using this namespace

ALL_DB_LINKS

ALL_DB_LINKS describes database links accessible to the user. This view does not display the PASSWORD column.

Related Views

- DBA_DB_LINKS describes all database links defined in the database. This view does not display the PASSWORD column.
- USER_DB_LINKS describes database links in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the database link
DB_LINK	VARCHAR2(12)	NOT NULL	Name of the database link
USERNAME	VARCHAR2(30)		Name of user when logging in
PASSWORD	VARCHAR2(30)		Login password for current user
HOST	VARCHAR2(200)		Net8 connect string
CREATED	DATE	NOT NULL	Creation time of the database link

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
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL / UNSUCCESSFUL

Column	Datatype	NULL	Description
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
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	VARCHAR2(3)		Auditing REFERENCES WHENEVER SUCCESSFUL / UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL / UNSUCCESSFUL

ALL_DEPENDENCIES

ALL_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers accessible to the 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(12)		Type of object
REFERENCED_OWNER	VARCHAR2(30)		Owner of the parent object
REFERENCED_NAME	VARCHAR2(64)		Type of parent object
REFERENCED_TYPE	VARCHAR2(12)		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
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Dependent column name
INFERRED	CHAR(1)		Whether inferred

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	Datatype	NULL	Description
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
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)		Whether the dimension is invalid (Y/N)

Column	Datatype	NULL	Description
REVISION	NUMBER		Dimension revision level

ALL_DIRECTORIES

ALL_DIRECTORIES describes all directories accessible to the 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
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object
SEQUENCE	NUMBER	NOT NULL	Sequence number, for ordering
LINE	NUMBER	NOT NULL	Line number at which this error occurs
POSITION	NUMBER	NOT NULL	Position in the line at which this error occurs
TEXT	VARCHAR2(4000)	NOT NULL	Text of the error

ALL_EVALUATION_CONTEXT_TABLES

ALL_EVALUATION_CONTEXT_TABLES describes all tables in all rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_TABLES describes all tables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_TABLES describes all tables in all 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 2-163
- "USER_EVALUATION_CONTEXT_TABLES" on page 2-247

ALL_EVALUATION_CONTEXT_VARS

ALL_EVALUATION_CONTEXT_VARS describes all variables in all rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_VARS describes all variables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_VARS describes all variables in all 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

See Also:

- "DBA_EVALUATION_CONTEXT_VARS" on page 2-163
- "USER_EVALUATION_CONTEXT_VARS" on page 2-248

ALL_EVALUATION_CONTEXTS

ALL_EVALUATION_CONTEXTS describes all 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 all 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 2-163
- "USER_EVALUATION_CONTEXTS" on page 2-248

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 2-165
- "USER_EXTERNAL_LOCATIONS" on page 2-248

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 (can be NULL)
DEFAULT_DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the default directory for the external table (can be NULL)
REJECT_LIMIT	NUMBER		Reject limit for the external table (default is 0)
ACCESS_TYPE	VARCHAR2(7)		Type of access parameters for the external table (CLOB/BLOB)
ACCESS_PARAMETERS	VARCHAR2(4000)		Access parameters for the external table

See Also:

- "DBA_EXTERNAL_TABLES" on page 2-165
- "USER_EXTERNAL_TABLES" on page 2-248

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"."atttribute"
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
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

Column	Datatype	NULL	Description
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)	Whether the partition belongs to a local index on a composite-partitioned table (YES | 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
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

Column	Datatype	Description
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 partition
COMPRESSION	VARCHAR2(8)	Indicates whether key compression is ENABLED or DISABLED for a partitioned index. 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. ■ 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)	The actual buffer pool for the partition
USER_STATS	VARCHAR2(3)	Whether the statistics were entered directly by the user (YES | 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)	

Column	Datatype	Description
PARAMETERS	VARCHAR2(1000)	

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

Column	Datatype	NULL	Description
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.
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ 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)		Whether the statistics were entered directly by the user
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 indexes on all tables accessible to the user. To gather statistics for this view and the related views DBA_INDEXES and USER_INDEXES, use the SQL ANALYZE statement.

> **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.

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.

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(12)		Type of index (NORMAL, BITMAP, FUNCTION-BASED NORMAL, FUNCTION-BASED BITMAP, or 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)		Whether the index is UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(11)		Whether the index is ENABLED or 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

Column	Datatype	NULL	Description
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. ■ 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.
STATUS	VARCHAR2(8)		Whether a nonpartitioned index is VALID or UNUSABLE
NUM_ROWS	NUMBER		Number of rows in the index

Column	Datatype	NULL	Description
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze the index
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)		Whether this index is partitioned (YES | NO)
TEMPORARY	VARCHAR2(1)		Whether the index is on a temporary table
GENERATED	VARCHAR2(1)		Whether the name of the index is system generated (Y|N)
SECONDARY	VARCHAR2(1)		Whether the index is a secondary object created by the ODCIIndexCreate method of the Oracle9*i* 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)		Whether the statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ 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
PCT_DIRECT_ACCESS	NUMBER		For a secondary index on an index-organized table, the percentage of rows with VALID guess
ITYP_OWNER	NUMBER		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 index as a whole (YES) or were estimated from statistics on underlying index partitions and subpartitions (NO).
DOMIDX_STATUS	VARCHAR		Reflects the status of the domain index ■ NULL: the specified index is not a domain index ■ VALID: the index is a valid domain index ■ IDXTYP_INVLD: the indextype of this domain index is invalid

Column	Datatype	NULL	Description
DOMIDX_OPSTATUS	VARCHAR		Reflects the status of an operation that was performed on a domain index ■ NULL: the specified index is not a domain index ■ VALID: the operation performed without errors ■ FAILED: the operation failed with an error
FUNCIDX_STATUS	VARCHAR		Indicates the status of a function-based index ■ NULL: this is not a function-based index ■ ENABLED: the function-based index is enabled ■ DISABLED: the function-based index is disabled

ALL_INDEXTYPE_COMMENTS

ALL_INDEXTYPE_COMMENTS lists all comments for user-defined indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_COMMENTS lists all comments for user-defined indextypes in the database.
- USER_INDEXTYPE_COMMENTS lists all comments for 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 2-167
- "USER_INDEXTYPE_COMMENTS" on page 2-249

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 2-167
- "USER_INDEXTYPE_OPERATORS" on page 2-250

ALL_INDEXTYPES

ALL_INDEXTYPES describes all indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPES describes all indextypes in the database.
- USER_INDEXTYPES describes all 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 indextype implementation (that is, containing the indextype operators)
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of indextype implementation type
INTERFACE_VERSION	NUMBER		Version of indextype interface
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of indextype implementation

Column	Datatype	NULL	Description
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype
PARTITIONING	VARCHAR2(10)		Kinds of local partitioning supported by the indextype

See Also:

- "DBA_INDEXTYPES" on page 2-167
- "USER_INDEXTYPES" on page 2-250

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.

Related Views

- DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database. This view does not display the OWNER_NAME column.
- 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
BASE_TYPE	VARCHAR2(7)		Base type of the type of the argument: ■ 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 2-168
- "USER_JAVA_ARGUMENTS" on page 2-250

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
OUTER	VARCHAR2(4000)		Outer class of this Java class if this Java class is an inner class

See Also:

- "DBA_JAVA_CLASSES" on page 2-168
- "USER_JAVA_CLASSES" on page 2-250

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 2-168
- "USER_JAVA_DERIVATIONS" on page 2-250

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
NAME	VARCHAR2(4000)		Name of the Java class

Column	Datatype	NULL	Description
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: ■ 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 volotile (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: ■ 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 2-168
- "USER_JAVA_FIELDS" on page 2-251

ALL_JAVA_IMPLEMENTS

ALL_JAVA_IMPLEMENTS describes interfaces implemented by the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_IMPLEMENTS describes interfaces implemented by all stored Java classes in the database.
- USER_JAVA_IMPLEMENTS 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
INTERFACE_NAME	VARCHAR2(4000)		Name of the interface identified by the INTERFACE_INDEX column

See Also:

- "DBA_JAVA_IMPLEMENTS" on page 2-168
- "USER_JAVA_IMPLEMENTS" on page 2-251

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: ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the referred inner class is declared static in the sorce file (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the referred inner class is declared final in the sorce file (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether the referred inner class is declared abstract in the sorce file (YES) or not (NO)
IS_INTERFACE	VARCHAR2(3)		Indicates whether the referred inner class is declared interface in the sorce file (YES) or not (NO)

See Also:

- "DBA_JAVA_INNERS" on page 2-169
- "USER_JAVA_INNERS" on page 2-251

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 2-169
- "USER_JAVA_LAYOUTS" on page 2-251

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

Column	Datatype	NULL	Description
METHOD_NAME	VARCHAR2(4000)		Name of the method identified by the METHOD_INDEX column
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the method: ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the method is a static method (YES) or not (NO)
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: ■ 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 2-169
- "USER_JAVA_METHODS" on page 2-252

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
LIBRARY	VARCHAR2(4000)		ncomp library shown in this row

See Also:

- "DBA_JAVA_NCOMPS" on page 2-169
- "USER_JAVA_NCOMPS" on page 2-252

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 2-170
- "USER_JAVA_RESOLVERS" on page 2-252

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 2-170
- "USER_JAVA_THROWS" on page 2-252

ALL_JOBS

ALL_JOBS describes all jobs in the database that are accessible to the current user.

See Also: *Oracle9i 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, Oracle 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

Column	Datatype	NULL	Description
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 this job has started and failed since its last success
WHAT	VARCHAR2(4000)		Body of the anonymous PL/SQL block that this job executes
NLS_ENV	VARCHAR2(4000)		Session parameters describing the NLS environment of the job
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 exiting 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

Column	Datatype	Description
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 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
TABLE_OWNER	VARCHAR2(30)		Owner of the table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(30)		Name of the LOB column
LOB_NAME	VARCHAR2(30)		Name of the partitioned LOB item
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
LOB_PARTITION_NAME	VARCHAR2(30)		Name of the LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of the corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
COMPOSITE	VARCHAR2(3)		Whether the partition is composite (YES | NO)
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER		Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(3)		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 is enabled for the LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size in bytes of the initial extent of the LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size in bytes of secondary extents of the LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for the LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(7)		Logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		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)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(30)		Name of the LOB column
LOB_NAME	VARCHAR2(30)		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		Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(3)		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)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size in bytes of the initial extent for the LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size in bytes of secondary extents for the LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of the LOB data partition

Column	Datatype	NULL	Description
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for the LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for the LOB data partition blocks

ALL_LOBS

ALL_LOBS displays the large objects (LOBs) contained in tables accessible to the 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
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

Column	Datatype	NULL	Description
FREEPOOLS	NUMBER		Number of freepools for this LOB segment
CACHE	VARCHAR2(10)		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
IN_ROW	VARCHAR2(3)		Indicates whether some of the LOBs are stored inline with the base row

Related Views

- DBA_OPERATOR_COMMENTS lists all comments for user-defined operators in the database.
- USER_OPERATOR_COMMENTS lists all comments for 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 2-184
- "USER_OPERATOR_COMMENTS" on page 2-257

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 with log group definition
COLUMN_NAME	VARCHAR2(4000)	NOT NULL	Name of the column or attribute of the object type column specified in the log group definition
POSITION	NUMBER		Original position of column or attribute in the definition of the object

ALL_LOG_GROUPS

ALL_LOG_GROUPS describes log group definitions on tables accessible to the current user.

Related Views

- DBA_LOG_GROUPS describes log group definitions on tables owned by the current users.
- USER_LOG_GROUPS describes log group definitions on all tables in the database.

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 with log group definition
ALWAYS	VARCHAR2(6)		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.

ALL_METHOD_PARAMS

ALL_METHOD_PARAMS describes the method parameters of all 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 all 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 2-179
- "USER_METHOD_PARAMS" on page 2-254

ALL_METHOD_RESULTS

ALL_METHOD_RESULTS describes the method results of all 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 all 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 2-179
- "USER_METHOD_RESULTS" on page 2-254

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 Oracle8*i* 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 2-72.
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 Corporation 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)

Column	Datatype	NULL	Description
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 in releases before 8.1 (or under pre-8.1 compatibility mode), Oracle attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, Oracle 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)
INCREFRESHTIM	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)
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	NOT NULL	The length (in bytes) of the query field

Column	Datatype	NULL	Description
QUERY	LONG	NOT NULL	SELECT expression of the materialized view definition
REVISION	NUMBER		Reserved for internal use

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 Oracle8*i* 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 Oracle8*i* 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
Note: The next 8 rows relate only to materialized join views and materialized aggregate views. They describe the two detail objects of a materialized view join.			
DETAILOBJ1_OWNER	VARCHAR2(30)	NOT NULL	The owner of the first object in the join
DETAILOBJ1_RELATION	VARCHAR2(30)	NOT NULL	The name of the first object in the join
DETAILOBJ1_COLUMN	VARCHAR2(30)	NOT NULL	The join column of the first object in the join
OPERATOR	CHAR(1)		The join operator
OPERATOR_TYPE	VARCHAR2(1)		Whether the join is an inner or outer join
DETAILOBJ2_OWNER	VARCHAR2(30)	NOT NULL	The owner of the second object in the join
DETAILOBJ2_RELATION	VARCHAR2(30)	NOT NULL	The name of the second object in the join
DETAILOBJ2_COLUMN	VARCHAR2(30)	NOT NULL	The join column of the second object in the 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 Oracle8*i* 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 Oracle8*i* and higher. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserted rows into the log.

Column	Datatype	NULL	Description
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 2-181
- "USER_MVIEW_LOGS" on page 2-255

ALL_MVIEW_REFRESH_TIMES

ALL_MVIEW_REFRESH_TIMES describes refresh times of materialized views accessible to the current owner.

Related Views

- DBA_MVIEW_REFRESH_TIMES describes refresh times of all materialized views in the database.
- USER_MVIEW_REFRESH_TIMES describes refresh times of all 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 2-181
- "USER_MVIEW_REFRESH_TIMES" on page 2-255

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 in releases before 8.1 (or under pre-8.1 compatibility mode), Oracle attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, then Oracle 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)
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)

Column	Datatype	NULL	Description
REWRITE_CAPABILITY	VARCHAR2(9)		Indicates whether the materialized view is eligible for rewrite, and if so, what rules must be followed: ■ 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 Oracle can apply any rewrite rule that is supported
REFRESH_MODE	VARCHAR2(6)		Refresh mode of the materialized view: ■ DEMAND - Oracle refreshes this materialized view whenever an appropriate refresh procedure is called ■ COMMIT - Oracle refreshes this materialized view when a transaction on one of the materialized view's masters commits ■ NEVER - Oracle never refreshes this materialized view
REFRESH_METHOD	VARCHAR2(8)		Default method used to refresh the materialized view (can be overridden through the API): ■ COMPLETE (C) - Materialized view is completely refreshed from the masters ■ FORCE (?) - Oracle performs a fast refresh if possible, otherwise a complete refresh ■ FAST (F) - Oracle performs an incremental refresh applying changes that correspond to changes in the masters since the last refresh ■ NEVER (N) - User specified that Oracle should not refresh this materialized view
BUILD_MODE	VARCHAR2(9)		Indicates how the materialized view was populated during creation: ■ 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 Oracle.

Column	Datatype	NULL	Description
FAST_REFRESHABLE	VARCHAR2(18)		Indicates whether the materialized view is eligible for incremental (fast) refresh. Oracle calculates this value statically, based on the materialized view definition query: ■ 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: ■ 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: ■ 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 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.

Column	Datatype	NULL	Description
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)
COMPILE_STATE	VARCHAR2(19)		Validity of the materialized view with respect to the objects upon which it depends: ■ 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.

See Also:

- "DBA_MVIEWS" on page 2-181
- "USER_MVIEWS" on page 2-256
- *Oracle9i Replication* for more information on materialized views to support replication
- *Oracle9i Data Warehousing Guide* for more information on materialized views to support data warehousing

ALL_NESTED_TABLES

ALL_NESTED_TABLES describes the nested tables in tables accessible to the 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 2-182
- "USER_OBJ_COLATTRS" on page 2-256

ALL_OBJECT_TABLES

ALL_OBJECT_TABLES describes all object tables accessible to the current user.

Related Views

- DBA_OBJECT_TABLES describes all object tables in the database.
- USER_OBJECT_TABLES describes all 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
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 entry belongs
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER		Minimum percentage of used space in a block
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size (in bytes) of the initial extent
NEXT_EXTENT	NUMBER		Size (in bytes) of secondary extents

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		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
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)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or 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)	NOT NULL	Owner of the type of the table
TABLE_TYPE	VARCHAR2(30)	NOT NULL	Type of the table
TEMPORARY	VARCHAR2(1)		Whether this is a temporary table

Column	Datatype	NULL	Description
SECONDARY	VARCHAR2(1)		Whether the object table is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge (Y | N)
NESTED	VARCHAR2(3)		Whether the table a nested table
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, 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)		Whether statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ 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 Oracle 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

ALL_OBJECTS

ALL_OBJECTS describes all objects accessible to the 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)

Column	Datatype	NULL	Description
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 (*object ID*) that Oracle assigns to row objects in object tables in the system.
OBJECT_TYPE	VARCHAR2(15)		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 command (including grants and revokes)
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 ODCIIndexCreate method of the Oracle9*i* 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

Column	Datatype	NULL	Description
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 operator bindings accessible to the current user.

Related Views

- DBA_OPBINDINGS describes bindings of all operators in the database.
- USER_OPBINDINGS describes bindings of all operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator

Column	Datatype	NULL	Description
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)		If the return type of the binding is an object type, the schema of the type
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT, SCAN CONTEXT, this column displays the schema of the implementation type used by the functional implementation of the operator as a scan context. (Blank if the operator was created without this syntax.) **See Also:** the CREATE OPERATOR statement in *Oracle9i SQL Reference*
IMPLEMENTATION_TYPE	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT, SCAN CONTEXT, this column displays the name of the implementation type used by the functional implementation of the operator as a scan context. (Blank if the operator was created without this syntax.) **See Also:** the CREATE OPERATOR statement in *Oracle9i SQL Reference*

ALL_OPERATOR_COMMENTS

ALL_OPERATOR_COMMENTS lists all comments for user-defined operators accessible to the current user.

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

Column	Datatype	NULL	Description
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 2-184
- "USER_OPERATORS" on page 2-257

ALL_OUTLINE_HINTS

ALL_OUTLINE_HINTS describes the set of hints stored in outlines accessible to the 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
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 this outline belongs
USED	VARCHAR2(9)		Whether the outline has ever been used (USED, UNUSED, or UNDEFINED)
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.

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
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
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
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO).
USER_STATS	VARCHAR2(3)		Whether the statistics were entered directly by the user
AVG_COL_LEN	NUMBER		Average length of the column, in bytes

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
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
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 provides object-level partitioning information for all partitioned indexes accessible to the current user.

Related Views

- DBA_PART_INDEXES provides such information for all partitioned indexes in the database.
- USER_PART_INDEXES provides such information for all 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 this partitioned index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		RANGE, HASH, SYSTEM, or UNKNOWN
SUBPARTITIONING_TYPE	VARCHAR2(7)		HASH, SYSTEM, or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this index
DEF_SUBPARTITION_COUNT	NUMBER	NOT NULL	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	NOT NULL	For a composite-partitioned index, the number of columns in the subpartitioning key
LOCALITY	VARCHAR2(6)		Whether this partitioned index is LOCAL or GLOBAL

Column	Datatype	NULL	Description
ALIGNMENT	VARCHAR2(12)		Whether this 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
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)		

ALL_PART_KEY_COLUMNS

ALL_PART_KEY_COLUMNS describes the partitioning key columns for partitioned objects accessible to the current user.

Related Views

- DBA_PART_KEY_COLUMNS describes partitioning key columns for all partitions of all partitioned objects in the database.
- USER_PART_KEY_COLUMNS describes partitioning key columns for partitions of 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 partitioned table or index
NAME	VARCHAR2(30)		Name of the partitioned table or index
OBJECT_TYPE	VARCHAR2(11)		Object type (TABLE | INDEX)
COLUMN_NAME	VARCHAR2(30)		Name of the column
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

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(30)	NOT NULL	Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB

Column	Datatype	NULL	Description
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(3)		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
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 provides object-level partitioning information for partitioned tables accessible to the current user.

Related Views

- DBA_PART_TABLES provides object-level partitioning information for all partitioned tables in the database.
- USER_PART_TABLES provides object-level partitioning information for all 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 (RANGE | HASH | SYSTEM | LIST)
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of subpartitioning (NONE | HASH | SYSTEM | LIST)
PARTITION_COUNT	NUMBER		Number of partitions in this 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
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.

Column	Datatype	NULL	Description
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)

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 all 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 all 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 2-186
- "USER_PENDING_CONV_TABLES" on page 2-259

ALL_POLICIES

ALL_POLICIES describes all security policies on all tables and views accessible to the current user.

Related Views

- DBA_POLICIES describes all policies in the database.
- USER_POLICIES describes policies on all tables and views owned by the current user. This view does not display the OBJECT_OWNER column.

See Also:

- *Oracle9i Database Concepts* for information on security policies and fine-grained access control
- The DBMS_RLS packages in *Oracle9i Supplied PL/SQL Packages and Types Reference* for information on administering security policies

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the table or view

Column	Datatype	NULL	Description
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)		If YES, policy is applied to query on the object
INS	VARCHAR2(3)		If YES, policy is applied to insert on the object
UPD	VARCHAR2(3)		If YES, policy is applied to update on the object
DEL	VARCHAR2(3)		If YES, policy is applied to delete on the object
CHK_OPTION	VARCHAR2(3)		Is check option enforced for this policy?
ENABLE	VARCHAR2(3)		Is this policy enabled?

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	Description
OWNER	VARCHAR2(30)	Owner of the procedure
OBJECT_NAME	VARCHAR2(30)	Name of the object: top-level function, procedure, or package name
PROCEDURE_NAME	VARCHAR2(30)	Name of the procedure
AGGREGATE	VARCHAR2(3)	(YES/NO) Identifies whether or not the procedure is an aggregate function
PIPELINED	VARCHAR2(3)	(YES/NO) Indicates whether or not the procedure is a pipelined table function
IMPLTYPEOWNER	VARCHAR2(30)	Name of the owner of the implementation type, if any
IMPLTYPENAME	VARCHAR2(30)	Name of the implementation type, if any

Column	Datatype	Description
PARALLEL	VARCHAR2(3)	(YES/NO) Indicates whether or not the procedure or function is parallel-enabled

ALL_PROPAGATION

ALL_PROPAGATION displays information about the Streams propagation jobs accessible to the current user.

Related View

- DBA_PROPAGATION displays information about all Streams propagation jobs 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 propagation source queue
SOURCE_QUEUE_NAME	VARCHAR2(30)		Name of the propagation source queue
DESTINATION_QUEUE_OWNER	VARCHAR2(30)		Owner of the propagation destination queue
DESTINATION_QUEUE_NAME	VARCHAR2(30)		Name of the propagation destination queue
DESTINATION_DBLINK	VARCHAR2(128)		Database link to access the propagation destination queue
RULE_SET_OWNER	VARCHAR2(30)		Propagation rule set owner
RULE_SET_NAME	VARCHAR2(30)		Propagation rule set name

See Also: "DBA_PROPAGATION" on page 2-188

ALL_PUBLISHED_COLUMNS

ALL_PUBLISHED_COLUMNS describes all existing source table columns for which users have privileges.

Related Views

- DBA_PUBLISHED_COLUMNS describes all existing source column tables.
- USER_PUBLISHED_COLUMNS describes all existing source table columns for which a user has privileges.

Column	Datatype	Description
CHANGE_SET_NAME	VARCHAR2(30)	Name of the change set
SOURCE_SCHEMA_NAME	VARCHAR2(30)	Table owner in the source system
SOURCE_TABLE_NAME	VARCHAR2(31)	Table name in the source system
PUB_ID	NUMBER	Publication identifier (change table object number)
COLUMN_NAME	VARCHAR2(30)	Column name
DATA_TYPE	VARCHAR2(30)	Column datatype
DATA_LENGTH	NUMBER	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	CHAR(1)	Nulls allowed (Y | N)?

ALL_QUEUE_TABLES

ALL_QUEUE_TABLES describes queues in the queue tables accessible to the current user.

Related Views

- DBA_QUEUE_TABLES describes queues in all queue tables in the database.
- USER_QUEUE_TABLES describes queues in the queue tables created in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema of the queue table
QUEUE_TABLE	VARCHAR2(30)		Name of the queue table
TYPE	VARCHAR2(7)		Type of user data: ■ RAW - Raw type ■ OBJECT - User-defined object type ■ VARIANT - Variant type (internal use only)
OBJECT_TYPE	VARCHAR2(61)		Object type of the payload when TYPE is OBJECT
SORT_ORDER	VARCHAR2(22)		User specified sort order
RECIPIENTS	VARCHAR2(8)		SINGLE or MULTIPLE recipients
MESSAGE_GROUPING	VARCHAR2(13)		NONE or TRANSACTIONAL

Column	Datatype	NULL	Description
COMPATIBLE	VARCHAR2(5)		Lowest release level which this queue table is compatible with (for example, 8.0.3)
PRIMARY_INSTANCE	NUMBER		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.
SECONDARY_INSTANCE	NUMBER		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.
OWNER_INSTANCE	NUMBER		Instance number of the instance which currently owns the queue table
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user
SECURE	VARCHAR2(3)		Indicates whether this queue table is secure (YES) or not (NO)

See Also: *Oracle9i Application Developer's Guide - Advanced Queuing* 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.

See Also: *Oracle9i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue

Column	Datatype	NULL	Description
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

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

Column	Datatype	NULL	Description
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
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
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

Column	Datatype	NULL	Description
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 summaries or 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(7)		MVIEW or SUMMARY
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.

Column	Datatype	NULL	Description
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
VERSION	VARCHAR2(26)		Oracle version of the materialized view **Note:** Oracle9*i* materialized views show ORACLE 8 MATERIALIZED VIEW.
QUERY_TXT	LONG		Query that defines the materialized view

See Also:

- "DBA_REGISTERED_MVIEWS" on page 2-192
- "USER_REGISTERED_MVIEWS" on page 2-261

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_RULE_SET_RULES

ALL_RULE_SET_RULES describes all rules in all rule sets accessible to the current user.

Related Views

- DBA_RULE_SET_RULES describes all rules in all rule sets in the database.
- USER_RULE_SET_RULES describes all rules in all rule sets owned by the current user. This view does not display the RULE_SET_OWNER column.

Column	Datatype	NULL	Description
RULE_SET_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule set

Column	Datatype	NULL	Description
RULE_SET_NAME	VARCHAR2(30)	NOT NULL	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_SET_RULE_ENABLED	VARCHAR2(8)		Indicates whether the rule is enabled in the rule set (ENABLED) or not (DISABLED)
RULE_SET_RULE_EVAL_CTX_OWNER	VARCHAR2(30)		Owner of the evaluation context specified when the rule was added to the rule set, if any
RULE_SET_RULE_EVAL_CTX_NAME	VARCHAR2(30)		Name of the evaluation context specified when the rule was added to the rule set, if any
RULE_SET_RULE_COMMENT	VARCHAR2(4000)		Comment specified when the rule was added to the rule set, if any

See Also:

- "DBA_RULE_SET_RULES" on page 2-202
- "USER_RULE_SET_RULES" on page 2-263

ALL_RULE_SETS

ALL_RULE_SETS describes all rule sets accessible to the current user.

Related Views

- DBA_RULE_SETS describes all rule sets in the database.
- USER_RULE_SETS describes all rule sets owned by the current user. This view does not display the RULE_SET_OWNER column.

Column	Datatype	NULL	Description
RULE_SET_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule set
RULE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the rule set
RULE_SET_EVAL_CONTEXT_OWNER	VARCHAR2(30)		Owner of the evaluation context associated with the rule set, if any
RULE_SET_EVAL_CONTEXT_NAME	VARCHAR2(30)		Name of the evaluation context associated with the rule set, if any
RULE_SET_COMMENT	VARCHAR2(4000)		Comment specified with the rule set, if any

See Also:

- "DBA_RULE_SETS" on page 2-202
- "USER_RULE_SETS" on page 2-263

ALL_RULES

ALL_RULES describes all rules accessible to the current user.

Related Views

- DBA_RULES describes all rules in the database.
- USER_RULES describes all rules owned by the current user. This view does not display the RULE_OWNER column.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_CONDITION	CLOB		Rule expression
RULE_EVALUATION_CONTEXT_OWNER	VARCHAR2(30)		Owner of the evaluation context associated with the rule, if any
RULE_EVALUATION_CONTEXT_NAME	VARCHAR2(30)		Name of the evaluation context associated with the rule, if any
RULE_ACTION_CONTEXT	RE$NV_LIST		Action context associated with the rule, if any
RULE_COMMENT	VARCHAR2(4000)		Comment specified with the rule, if any

See Also:

- "DBA_RULES" on page 2-202
- "USER_RULES" on page 2-263

ALL_SEQUENCES

ALL_SEQUENCES describes all sequences accessible to the 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_SOURCE

ALL_SOURCE describes the text source of all stored objects accessible to the user.

Related Views

- DBA_SOURCE describes the text source of all stored objects in the database.
- USER_SOURCE describes the text source of all 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 enables publishers to see all existing source tables. You must have the SELECT_CATALOG_ROLE privilege to access this view.

Related Views

- DBA_SOURCE_TABLES describes all existing source tables.
- USER_SOURCE_TABLES describes all existing source tables to which the user has subscribe privileges.

Column	Datatype	Description
SOURCE_SCHEMA_NAME	VARCHAR2(30)	The table owner in the source system
SOURCE_TABLE_NAME	VARCHAR2(31)	The table name in the source system

ALL_SQLJ_TYPE_ATTRS

ALL_SQLJ_TYPE_ATTRS describes the attributes of all 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 all 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.

Column	Datatype	NULL	Description
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 2-204
- "USER_SQLJ_TYPE_ATTRS" on page 2-264

ALL_SQLJ_TYPE_METHODS

ALL_SQLJ_TYPE_METHODS describes the methods of all 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 all SQLJ 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
EXTERNAL_VAR_NAME	VARCHAR2(4000)		Name of the external variable
METHOD_NO	NUMBER	NOT NULL	Method number that distinguishes overloaded method (not to be used as an ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method

Column	Datatype	NULL	Description
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)
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_SQLJ_TYPE_METHODS" on page 2-204
- "USER_SQLJ_TYPE_METHODS" on page 2-264

ALL_SQLJ_TYPES

ALL_SQLJ_TYPES describes all 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 all SQLJ 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
EXTERNAL_NAME	VARCHAR2(4000)		External class name of the type
USING	VARCHAR2(21)		Representation 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)

Column	Datatype	NULL	Description
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

See Also:

- "DBA_SQLJ_TYPES" on page 2-204
- "USER_SQLJ_TYPES" on page 2-264

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

Column	Datatype	NULL	Description
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.

ALL_STREAMS_GLOBAL_RULES

ALL_STREAMS_GLOBAL_RULES displays information about global rules created for the Streams capture/apply/propagation processes that interact with queues accessible to the current user.

Related View

- DBA_STREAMS_GLOBAL_RULES displays information about all global rules created by Streams capture/apply/propagation processes.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process: capture/propagation/apply process
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process (CAPTURE, PROPAGATION, or APPLY)
RULE_TYPE	VARCHAR2(7)		Type of rule: DML or DDL
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether to include tagged LCR (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Name of the database where the LCRs originated
RULE_NAME	VARCHAR2(30)		Name of the rule to be applied
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		Generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_GLOBAL_RULES" on page 2-205

ALL_STREAMS_SCHEMA_RULES

ALL_STREAMS_SCHEMA_RULES displays information about rules created by the Streams capture / apply / propagation processes for all user's schemas.

Related View

- DBA_STREAMS_SCHEMA_RULES displays information about all schema rules created by Streams capture / apply / propagation processes.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process: capture / propagation / apply process
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process (CAPTURE, PROPAGATION, or APPLY)
SCHEMA_NAME	VARCHAR2(30)		Name of the schema selected by this rule
RULE_TYPE	VARCHAR2(7)		Type of rule: DML or DDL
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether to include tagged LCR (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Name of the database where the LCRs originated
RULE_NAME	VARCHAR2(30)		Name of the rule to be applied
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		Generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_SCHEMA_RULES" on page 2-205

ALL_STREAMS_TABLE_RULES

ALL_STREAMS_TABLE_RULES displays information about rules created by Streams processes and jobs on the tables accessible to the current user.

Related View

- DBA_STREAMS_TABLE_RULES displays information about rules created by Streams processes and jobs on all tables in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process: capture / propagation / apply process

Column	Datatype	NULL	Description
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process (CAPTURE, PROPAGATION, or APPLY)
TABLE_OWNER	VARCHAR2(30)		Owner of the table selected by this rule
TABLE_NAME	VARCHAR2(30)		Name of the table selected by this rule
RULE_TYPE	VARCHAR2(7)		Type of rule: DML or DDL
DML_CONDITION	VARCHAR2(4000)		Row subsetting condition
SUBSETTING_OPERATION	VARCHAR2(6)		DML operation for row subsetting: ■ INSERT ■ UPDATE ■ DELETE
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether to include tagged LCR (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Name of the database where the LCRs originated
RULE_NAME	VARCHAR2(30)		Name of the rule to be applied
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		Generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_TABLE_RULES" on page 2-205

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
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name

Column	Datatype	NULL	Description
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
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
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
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 gathered on partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

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

ALL_SUBSCRIBED_COLUMNS

ALL_SUBSCRIBED_COLUMNS enables publishers to view the columns of published tables to which they have subscribed.

Related Views

- DBA_SUBSCRIBED_COLUMNS enables publishers to view the columns of published tables to which they have subscribed. You must have the SELECT_CATALOG_ROLE privilege to access this view.
- USER_SUBSCRIBED_COLUMNS enables publishers to view the columns of published tables to which they have subscribed.

Column	Datatype	Description
HANDLE	NUMBER	The subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	The source table schema identifier
SOURCE_TABLE_NAME	VARCHAR2(31)	The source table identifier
COLUMN_NAME	VARCHAR2(30)	The source table column identifier

ALL_SUBSCRIBED_TABLES

ALL_SUBSCRIBED_TABLES enables subscribers to view all published tables to which they have subscribed.

Related Views

- DBA_SUBSCRIBED_TABLES enables publishers to view all published tables to which they have subscribed.
- USER_SUBSCRIBED_TABLES enables subscribers to view all published tables to which they have subscribed.

Column	Datatype	Description
HANDLE	NUMBER	The subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	The source table schema identifier
SOURCE_TABLE_NAME	VARCHAR2(31)	The source table identifier
VIEW_NAME	VARCHAR2(30)	Holds the generated view name so that change data capture can drop the view when necessary
CHANGE_SET_NAME	VARCHAR2(30)	The name of the change set to which the change table belongs

ALL_SUBSCRIPTIONS

ALL_SUBSCRIPTIONS enables subscribers to view all subscriptions. Neither ALL_SUBSCRIPTIONS nor USER_SUBSCRIPTIONS contains the USERNAME column.

Related Views

- DBA_SUBSCRIPTIONS enables publishers to view all subscriptions. You must have the SELECT_CATALOG_ROLE privilege to access this view. DBA_SUBSCRIPTIONS contains the USERNAME column, which lists the user name of the subscriber.
- USER_SUBSCRIPTIONS enables subscribers to view all of their subscriptions.

Column	Datatype	Description
HANDLE	NUMBER	The subscription handler
SET_NAME	VARCHAR2(30)	The change set identifier
CREATED	DATE	Subscription creation date
STATUS	VARCHAR2(1)	N indicates not yet active, A indicates is currently active
EARLIEST_SCN	NUMBER	The starting point for the select / view
LATEST_SCN	NUMBER	The ending point for the select / view
DESCRIPTION	VARCHAR2(30)	A comment field for the subscriber
LAST_PURGED	DATE	The last time the subscriber called the PURGE_WINDOW for this subscription
LAST_EXTENDED	DATE	The last time the subscriber called the EXTEND_WINDOW for this subscription

ALL_SUMDELTA

ALL_SUMDELTA lists direct path load entries accessible to the user.

Column	Datatype	NULL	Description
TABLEOBJ#	NUMBER	NOT NULL	Object number of the table
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 all synonyms accessible to the user.

Related Views

- DBA_SYNONYMS describes all synonyms in the database.
- USER_SYNONYMS describes all 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

ALL_TAB_COL_STATISTICS

ALL_TAB_COL_STATISTICS provides column statistics and histogram information extracted from "ALL_TAB_COLUMNS" on page 2-120.

Related Views

- DBA_TAB_COL_STATISTICS provides such information extracted from "DBA_TAB_COLUMNS" on page 2-207.
- USER_TAB_COL_STATISTICS provides such information extracted from "USER_TAB_COLUMNS" on page 2-266.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column

Column	Datatype	NULL	Description
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)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

ALL_TAB_COLUMNS

ALL_TAB_COLUMNS describes the columns of all tables, views, and clusters accessible to the user. To gather statistics for this view, use the SQL ANALYZE 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 all 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	Table, view, or cluster name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(30)		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

Column	Datatype	NULL	Description
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	NOT NULL	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		These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views. This view now picks up these values from HIST_HEAD$ rather than COL$.
LOW_VALUE	RAW(32)		
HIGH_VALUE	RAW(32)		
DENSITY	NUMBER		
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		The number of buckets in histogram for the column **Note:** The number of buckets in a histogram is specified in the SIZE parameter of the SQL statement ANALYZE. However, Oracle 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, Oracle 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		The date on which this column was most recently analyzed
SAMPLE_SIZE			The sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		The 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)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

Column	Datatype	NULL	Description
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: ■ 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: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2

ALL_TAB_COMMENTS

ALL_TAB_COMMENTS describes comments on tables and views accessible to the user.

Related Views

- DBA_TAB_COMMENTS describes comments on all tables and views in the database.
- USER_TAB_COMMENTS describes comments on all 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
TABLE_TYPE	VARCHAR2(11)		Type of the object
COMMENTS	VARCHAR2(4000)		Comment on the object

ALL_TAB_HISTOGRAMS

ALL_TAB_HISTOGRAMS describes histograms on tables and views accessible to the 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
OWNER	VARCHAR2(30)		Owner of table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
ENDPOINT_NUMBER	NUMBER		Histogram bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint value for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		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, Oracle 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
TRUNCATED	VARCHAR2(3)		Whether or not the table was truncated since the last time statistics were gathered

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

Column	Datatype	NULL	Description
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
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)		User statistics

ALL_TAB_PRIVS

ALL_TAB_PRIVS lists the grants on objects where the user or PUBLIC is the grantee. This view does not display the OWNER column.

Related Views

- DBA_TAB_PRIVS lists all grants on all objects in the database. This view does not display the TABLE_SCHEMA column.
- USER_TAB_PRIVS lists grants on all objects where the current user is the grantee. This view does not display the TABLE_SCHEMA column.

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 is granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_PRIVS_MADE

ALL_TAB_PRIVS_MADE lists the all object grants made by the current user or made on the objects owned by the current user.

Related View

- USER_TAB_PRIVS_MADE lists grants on all objects owned by the current user. 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

Column	Datatype	NULL	Description
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_PRIVS_RECD

ALL_TAB_PRIVS_RECD lists object grants for which the user or PUBLIC is the grantee.

Related View

- USER_TAB_PRIVS_RECD lists object grants for which the current user is the grantee. 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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_SUBPARTITIONS

ALL_TAB_SUBPARTITIONS describes, for each table subpartition accessible to the 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
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)		Were the statistics entered directly by the user?

ALL_TABLES

ALL_TABLES describes all relational tables accessible to the user. To gather statistics for this view, use the SQL ANALYZE statement.

Related Views

- DBA_TABLES describes all relational tables in the database.
- USER_TABLES describes all 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)		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 entry belongs. If IOT_TYPE column is not null, 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 the secondary extension bytes; NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables

Column	Datatype	NULL	Description
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 this segment; NULL for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this 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		The average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		The number of blocks on the freelist
DEGREE	VARCHAR2(10)		The number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		The number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or 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.

Column	Datatype	NULL	Description
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 Oracle9*i* Data Cartridge (Y | N)
NESTED	VARCHAR2(3)		Is the table a nested table?
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
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)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ 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 Oracle 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

See Also:

- "DBA_TABLES" on page 2-209
- "USER_TABLES" on page 2-268

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)	NOT NULL	Owner of the triggers
TRIGGER_NAME	VARCHAR2(30)	NOT NULL	Name of the trigger
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
TABLE_NAME	VARCHAR2(30)	NOT NULL	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_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)	NOT NULL	Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)	NOT NULL	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(216)		The DML, DDL, or database event that fires the trigger. For a listing of triggering events, see the CREATE TRIGGER statement in *Oracle9i SQL Reference*.
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined

Column	Datatype	NULL	Description
BASE_OBJECT_TYPE	VARCHAR2(16)		The 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 columns indicates the table/view name on which the trigger is defined
COLUMN_NAME	VARCHAR2(30)		Name of the nested table column (if nested table trigger), else null
REFERENCING_NAMES	VARCHAR2(87)		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)		The action type of the trigger body (CALL or PL/SQL)
TRIGGER_BODY	LONG		Statement(s) executed by the trigger when it fires

ALL_TYPE_ATTRS

ALL_TYPE_ATTRS describes the attributes of all 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 all 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
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

Column	Datatype	NULL	Description
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_TYPE_ATTRS" on page 2-212
- "USER_TYPE_ATTRS" on page 2-269

ALL_TYPE_METHODS

ALL_TYPE_METHODS describes the methods of all 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 all 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

Column	Datatype	NULL	Description
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)
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 2-212
- "USER_TYPE_METHODS" on page 2-269

ALL_TYPE_VERSIONS

ALL_TYPE_VERSIONS describes all versions of all object types accessible to the current user.

Related Views

- DBA_TYPE_VERSIONS describes all versions of all object types in the database.
- USER_TYPE_VERSIONS describes all versions of all 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
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 2-213
- "USER_TYPE_VERSIONS" on page 2-269

ALL_TYPES

ALL_TYPES describes all object types accessible to the current user.

Related Views

- DBA_TYPES describes all object types in the database.
- USER_TYPES describes all 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 2-213
- "USER_TYPES" on page 2-269

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

Column	Datatype	NULL	Description
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 in the database, and contains more columns than ALL_USERS. See "DBA_USERS" on page 2-214.
- USER_USERS describes the current user, and contains more columns that ALL_USERS. See "USER_USERS" on page 2-270.

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

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
OBJECT_TYPE	VARCHAR2(6)		Indicates the type of object for which statistics have been collected: COLUMN or INDEX

Column	Datatype	NULL	Description
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 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

Column	Datatype	NULL	Description
ELEMENT_SUBSTITUTABLE	VARCHAR2(25)		Indicates whether the varray element is substitutable (Y) or not (N)

ALL_VIEWS

ALL_VIEWS describes all views accessible to the current user.

Related Views

- DBA_VIEWS describes all views in the database.
- USER_VIEWS describes all 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

AUDIT_ACTIONS

AUDIT_ACTIONS describes audit trail action type codes.

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 3–3 on page 3-126 for a listing of the action type codes
NAME	VARCHAR2(27)	NOT NULL	Name of the type of audit trail action

CATALOG

CATALOG is included for compatibility with Oracle version 5. Oracle Corporation recommends that you not use this view.

CAT

CAT is a synonym for USER_CATALOG.

See Also: "USER_CATALOG" on page 2-244

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_SOURCES

CHANGE_SOURCES enables publishers to view existing change sources. You must have the SELECT_CATALOG_ROLE privilege to access this view.

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		The database identifier of the Oracle host instance (set by CDC when the first log is added to LogMiner)
LOG_DIRECTORY	VARCHAR2(4000)	NOT NULL	File system path on the CDC machine where log files for this source are maintained
LOGFILE_PATTERN	VARCHAR2(30)		This pattern enables you to recognize log files (for example, *.log) if the log location has files in it other than log files
SOURCE_DESCRIPTION	VARCHAR2(255)		Comment field for customers
CREATED	DATE	NOT NULL	Creation date of the change source

CHANGE_SETS

CHANGE_SETS enables publishers to view existing change sets. You must have the SELECT_CATALOG_ROLE privilege to access this view.

Column	Datatype	NULL	Description
SET_NAME	VARCHAR2(30)	NOT NULL	User specified identifier. For synchronous change sources, there is one pre-created source with the SET_NAME "SYNC_SET."
CHANGE_SOURCE_NAME	VARCHAR2(30)	NOT NULL	The change source to which this set belongs
BEGIN_DATE	DATE		The starting point for capturing change data
END_DATE	DATE		The stopping point for capturing change data
BEGIN_SCN	NUMBER		The starting point for capturing change data
END_SCN	NUMBER		The stopping point for capturing change data
FRESHNESS_DATE	DATE		The stopping point for the last successful advance
FRESHNESS_SCN	NUMBER		The stopping point for the last successful advance
ADVANCE_ENABLED	VARCHAR2(1)	NOT NULL	Identifies eligibility for advance (Y|N)
IGNORE_DDL	VARCHAR2(1)	NOT NULL	Identifies whether to continue (Y) or stop (N)
CREATED	DATE	NOT NULL	The creation date of the change set

Column	Datatype	NULL	Description
ROLLBACK_SEGMENT_NAME	VARCHAR2(30)		Used during advance (optional)
ADVANCING	VARCHAR2(1)	NOT NULL	Identifies whether or not (Y|N) an advance is in progress for this set (set by advance)
PURGING	VARCHAR2(1)	NOT NULL	Identifies whether or not (Y|N) a purge of old data is in progress for this change set (set by advance)
LOWEST_SCN	NUMBER	NOT NULL	Current low water mark for change data remaining in this set after the latest purge
TABLESPACE	VARCHAR2(30)	NOT NULL	Location where advance creates LCR staging tables

CHANGE_TABLES

CHANGE_TABLES enables publishers to see existing change tables. You must have the SELECT_CATALOG_ROLE privilege to access this view.

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		The system commit SCN when this table was created
CAPTURED_VALUES	VARCHAR2(1)	NOT NULL	Indicates the values O (old), N (new), or B (both)

CLU

CLU is a synonym for USER_CLUSTERS.

See Also: "USER_CLUSTERS" on page 2-244

CODE_PIECES

Oracle accesses this view to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views.

See Also: "DBA_OBJECT_SIZE" on page 2-183 and "USER_OBJECT_SIZE" on page 2-256

CODE_SIZE

Oracle accesses this view to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views.

See Also: "DBA_OBJECT_SIZE" on page 2-183 and "USER_OBJECT_SIZE" on page 2-256

COL

COL is included for compatibility with Oracle version 5. Oracle Corporation recommends that you not use this view.

COLS

COLS is a synonym for USER_TAB_COLUMNS.

See Also: "USER_TAB_COLUMNS" on page 2-266

COLUMN_PRIVILEGES

COLUMN_PRIVILEGES is included for compatibility with Oracle version 6. Oracle Corporation does not recommend use of this view.

DATABASE_PROPERTIES

DATABASE_PROPERTIES lists the current default temporary tablespace name.

Column	Datatype	NULL	Description
PROPERTY_NAME	VARCHAR2(30)	NOT NULL	Property name of the default temporary tablespace
PROPERTY_VALUE	VARCHAR2(4000)		Property value of the default temporary tablespace
DESCRIPTION	VARCHAR2(4000)		Property description of the default temporary tablespace

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)		The database ID at the other end of the connection
SESS#	NUMBER		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(2000)		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

Column	Datatype	NULL	Description
OS_USER	VARCHAR2(2000)		Time automatic recovery (RECO) last tried to recover the transaction
OS_TERMINAL	VARCHAR2(2000)		Time automatic recovery (RECO) last tried to recover the transaction
HOST	VARCHAR2(2000)		Name of the host machine for the end-user
DB_USER	VARCHAR2(30)		Name of the host machine for the end-user
COMMIT#	VARCHAR2(16)		Name of the host machine for the end-user

DBA_ALL_TABLES

DBA_ALL_TABLES displays descriptions of all object tables and relational tables in the database. Its columns are the same as those in "ALL_ALL_TABLES" on page 2-6.

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 2-8

DBA_APPLY_CONFLICT_COLUMNS

DBA_APPLY_CONFLICT_COLUMNS displays information about conflict resolution 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 2-9

DBA_APPLY_DML_HANDLERS

DBA_APPLY_DML_HANDLERS displays information about the DML handler 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 2-9

DBA_APPLY_ERROR

DBA_APPLY_ERROR displays information about error transactions. Its columns are the same as those in ALL_APPLY_ERROR.

See Also: "ALL_APPLY_ERROR" on page 2-10

DBA_APPLY_INSTANTIATED_OBJECTS

DBA_APPLY_INSTANTIATED_OBJECTS displays information about objects instantiated.

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	NUMBER	NOT NULL	Type of the object at the source database
INSTANTIATION_SCN	NUMBER		Point in time when the object was instantiated at the source database
IGNORE_SCN	NUMBER		SCN lower bound for messages that will be considered for apply
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes will be applied

DBA_APPLY_KEY_COLUMNS

DBA_APPLY_KEY_COLUMNS displays information about alternative key columns for all STREAMS tables in the database. Its columns are the same as those in ALL_APPLY_KEY_COLUMNS.

See Also: "ALL_APPLY_KEY_COLUMNS" on page 2-11

DBA_APPLY_PARAMETERS

DBA_APPLY_PARAMETERS displays information about all parameters for the apply process. Its columns are the same as those in ALL_APPLY_PARAMETERS.

See Also: "ALL_APPLY_PARAMETERS" on page 2-11

DBA_APPLY_PROGRESS

DBA_APPLY_PROGRESS displays progress information 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 2-12

DBA_ASSOCIATIONS

DBA_ASSOCIATIONS describes all user-defined statistics in the database. Its columns are the same as those for "ALL_ASSOCIATIONS" on page 2-14.

DBA_AUDIT_EXISTS

DBA_AUDIT_EXISTS lists audit trail entries produced by 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(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry
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(27)		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

Column	Datatype	NULL	Description
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
GRANTEE	VARCHAR2(30)		Name of 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 server message code generated by the action. Some useful values: ■ 0: the 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

DBA_AUDIT_OBJECT

DBA_AUDIT_OBJECT contains audit trail records for all objects in the system.

Related View

- USER_AUDIT_OBJECT contains 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
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement

Column	Datatype	NULL	Description
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(27)		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 server message code generated by the action. Some useful values: ■ 0: the 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

DBA_AUDIT_SESSION

DBA_AUDIT_SESSION lists 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
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(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
LOGOFF_TIME	DATE		Timestamp for 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
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
RETURNCODE	NUMBER	NOT NULL	Oracle server message code generated by the action. Some useful values: ■ 0: the 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

DBA_AUDIT_STATEMENT

DBA_AUDIT_STATEMENT lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements throughout the database.

Related View

- USER_AUDIT_STATEMENT contains 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(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of object affected by the action
ACTION_NAME	VARCHAR2(27)		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 ADMIN option
GRANTEE	VARCHAR2(30)		Name of grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
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

Column	Datatype	NULL	Description
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ 0: the 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

DBA_AUDIT_TRAIL

DBA_AUDIT_TRAIL lists all audit trail entries.

Related View

- USER_AUDIT_TRAIL contains 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(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
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 type code corresponding to the action
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column
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
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement

Column	Datatype	NULL	Description
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 ADMIN option
GRANTEE	VARCHAR2(30)		Name of grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
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).
LOGOFF_TIME	DATE		Timestamp for 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: ■ DATABASE: authentication was done by password ■ NETWORK: authentication was done by Net8 or the Advanced Security option ■ PROXY: the client was authenticated by another user; the name of the proxy user follows the method type
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 server message code generated by the action. Some useful values: ■ 0: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

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

DBA_AW_PS

DBA_AW_PS describes the pagespaces in all analytic workspaces in the database.

Related View

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
OWNER	VARCHAR2(30)	NOT NULL	Owner of the analytic workspace
AW_NUMBER	NUMBER	NOT NULL	Number of the analytic workspace
AW_NAME	VARCHAR2(30)		Name of the analytic workspace
PSNUMBER	NUMBER(10)		Number of the pagespace
GENERATIONS	NUMBER		Number of active generations in the pagespace

See Also:

- "USER_AW_PS" on page 2-243
- *Oracle9i OLAP User's Guide* for more information about this view

DBA_AWS

DBA_AWS describes all analytic workspaces in the database.

Related View

USER_AWS describes the analytic workspaces 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 analytic workspace
AW_NUMBER	NUMBER	NOT NULL	Number of the analytic workspace

Column	Datatype	NULL	Description
AW_NAME	VARCHAR2(30)		Name of the analytic workspace
PAGESPACES	NUMBER		Number of pagespaces in the analytic workspace
GENERATIONS	NUMBER		Number of active generations in the analytic workspace

See Also:

- "USER_AWS" on page 2-243
- *Oracle9i OLAP User's Guide* for more information about this view

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 2-15

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 2-15

DBA_CAPTURE_PARAMETERS

DBA_CAPTURE_PARAMETERS displays information about all parameters for the capture process. Its columns are the same as those in ALL_CAPTURE_PARAMETERS.

See Also: "ALL_CAPTURE_PARAMETERS" on page 2-16

DBA_CAPTURE_PREPARED_DATABASE

DBA_CAPTURE_PREPARED_DATABASE displays information about whether the local database is prepared for instantiation. Its columns are the same as those in ALL_CAPTURE_PREPARED_DATABASE.

See Also: "ALL_CAPTURE_PREPARED_DATABASE" on page 2-17

DBA_CAPTURE_PREPARED_SCHEMAS

DBA_CAPTURE_PREPARED_SCHEMAS displays information about all schemas at the local database that are prepared for instantiation. Its columns are the same as those in ALL_CAPTURE_PREPARED_SCHEMAS.

See Also: "ALL_CAPTURE_PREPARED_SCHEMAS" on page 2-17

DBA_CAPTURE_PREPARED_TABLES

DBA_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation. Its columns are the same as those in ALL_CAPTURE_PREPARED_TABLES.

See Also: "ALL_CAPTURE_PREPARED_TABLES" on page 2-17

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 2-18.

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 2-18.

DBA_CLUSTERS

DBA_CLUSTERS describes all clusters in the database. Its columns are the same as those in "ALL_CLUSTERS" on page 2-19.

DBA_COL_COMMENTS

DBA_COL_COMMENTS describes comments on columns of all tables and views in the database. Its columns are the same as those in "ALL_COL_COMMENTS" on page 2-20.

DBA_COL_PRIVS

DBA_COL_PRIVS describes all object column grants in the database. Its columns are the same as those in "ALL_COL_PRIVS" on page 2-21.

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 2-22.

DBA_CONS_COLUMNS

DBA_CONS_COLUMNS describes all columns in the database that are specified in constraint definitions. Its columns are the same as those in "ALL_CONS_COLUMNS" on page 2-23.

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	ID 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

Column	Datatype	NULL	Description
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_DB_LINKS

DBA_DB_LINKS describes all database links in the database. Its columns are the same as those in "ALL_DB_LINKS" on page 2-27.

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
TYPE	VARCHAR2(40)		Lock type: CURSOR, TABLE/PROCEDURE/TYPE, BODY, TRIGGER, INDEX, CLUSTER
MODE_HELD	VARCHAR2(9)		Lock mode: NONE, NULL, SHARE, EXCLUSIVE
MODE_REQUESTED	VARCHAR2(9)		Lock request type: 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 2-29.

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 2-30.

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 2-31.

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 2-31.

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 2-32.

DBA_DIMENSIONS

DBA_DIMENSIONS represents dimension objects. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-32.

DBA_DIRECTORIES

DBA_DIRECTORIES describes all directory objects in the database. Its columns are the same as those in "ALL_DIRECTORIES" on page 2-33.

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: ■ 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: ■ 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_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 2-33.

DBA_EVALUATION_CONTEXT_TABLES

DBA_EVALUATION_CONTEXT_TABLES describes all 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 2-34

DBA_EVALUATION_CONTEXT_VARS

DBA_EVALUATION_CONTEXT_VARS describes all 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 2-34

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 2-35

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
FILE_NAME	VARCHAR2(100)	NOT NULL	Name of the export file
USER_NAME	VARCHAR2(30)	NOT NULL	Name of user who executed export

Column	Datatype	NULL	Description
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 all segments in the database.

Related View

- USER_EXTENTS describes extents in segments associated with objects owned by the current user. 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
SEGMENT_TYPE	VARCHAR2(17)		Type of the segment: INDEX PARTITION, TABLE PARTITION

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent
EXTENT_ID	NUMBER	NOT NULL	Extent number in the segment
FILE_ID	NUMBER	NOT NULL	Name of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER	NOT NULL	Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the first extent block
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)

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 2-36

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 2-36

DBA_FREE_SPACE

DBA_FREE_SPACE lists the free extents in all tablespaces.

Related View

- USER_FREE_SPACE lists the free extents in tablespaces accessible to the current user.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the extent
FILE_ID	NUMBER	NOT NULL	ID number of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent

Column	Datatype	NULL	Description
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER	NOT NULL	Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the first extent block

DBA_FREE_SPACE_COALESCED

DBA_FREE_SPACE_COALESCED contains statistics on coalesced space in tablespaces.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of tablespace
TOTAL_EXTENTS	NUMBER		Total number of free extents in tablespace
EXTENTS_COALESCED	NUMBER		Total number of coalesced free extents in tablespace
PERCENT_EXTENTS _COALESCED	NUMBER		Percentage of coalesced free extents in tablespace
TOTAL_BYTES	NUMBER		Total number of free bytes in tablespace
BYTES_COALESCED	NUMBER		Total number of coalesced free bytes in tablespace
TOTAL_BLOCKS	NUMBER		Total number of free Oracle blocks in tablespace
BLOCKS_COALESCED	NUMBER		Total number of coalesced free Oracle blocks in tablespace
PERCENT_BLOCKS _COALESCED	NUMBER		Percentage of coalesced free Oracle blocks in tablespace

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 2-37.

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 2-38.

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 2-39.

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 2-41.

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 2-43.

DBA_INDEXTYPE_COMMENTS

DBA_INDEXTYPE_COMMENTS lists all comments for user-defined indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_COMMENTS.

See Also: "ALL_INDEXTYPE_COMMENTS" on page 2-46

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 2-46

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 2-47

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 2-48.

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 2-48

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 2-49

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 2-50

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 2-51

DBA_JAVA_IMPLEMENTS

DBA_JAVA_IMPLEMENTS describes interfaces implemented by all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_IMPLEMENTS.

See Also: "ALL_JAVA_IMPLEMENTS" on page 2-53

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 2-53

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 2-54

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 2-55

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 2-57

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 this 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, java.io.FilePermission
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 2-252

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 2-57

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 2-58

DBA_JOBS

DBA_JOBS describes all jobs in the database. Its columns are the same as those in "ALL_JOBS" on page 2-59.

See Also: *Oracle9i 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".
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 2-60.

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

Column	Datatype	NULL	Description
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 2-61.

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 2-61.

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 2-63.

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 2-64.

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

Column	Datatype	NULL	Description
LOCK_TYPE	VARCHAR2(26)		Lock type **See Also:** For a listing of lock types, see Appendix B, "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 B, "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 2-172

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 2-65.

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 2-66.

DBA_LOGSTDBY_EVENTS

DBA_LOGSTDBY_EVENTS contains 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 logs.

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 shows the logs registered for a logical standby database.

Column	Datatype	NULL	Description
THREAD#	NUMBER	NOT NULL	Thread ID of the archived redo log. The THREAD number is 1 for a single instance. For Oracle9*i* Real Application Clusters, this column will contain different numbers.
SEQUENCE#	NUMBER	NOT NULL	Sequence number of the archived redo log file
FIRST_CHANGE#	NUMBER		SCN of the current archived redo log
NEXT_CHANGE#	NUMBER		SCN of the next archived redo log
FIRST_TIME	DATE		Date of the current archived redo log
NEXT_TIME	DATE		Date of the next archived redo log
FILE_NAME	VARCHAR2(513)		Name of the archived redo log
TIMESTAMP	DATE		Time when the archived redo log was registered
DICT_BEGIN	VARCHAR2(3)		(YES/NO) Indicates whether the beginning of the dictionary build is in this particular archived redo log
DICT_END	VARCHAR2(3)		(YES/NO) Indicates whether the end of the dictionary build is in this particular archived redo log

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 identifies tables that have no primary and no non-null unique indexes. Most of the tables displayed in 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)	NOT NULL	Schema name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

Column	Datatype	NULL	Description
BAD_COLUMN	VARCHAR2(1)		■ Y - the 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 column, 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 contains the list of parameters used by the log apply services for logical standby databases.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the parameter: ■ 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 table ■ 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 applying 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.
VALUE	VARCHAR2(2000)		Value of the parameter

DBA_LOGSTDBY_PROGRESS

DBA_LOGSTDBY_PROGRESS describes the progress of log apply services on the logical standby database.

Column	Datatype	NULL	Description
APPLIED_SCN	NUMBER		Shows the 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.
APPLIED_TIME	DATE		Estimate of the time and date of the APPLIED_SCN
READ_SCN	NUMBER		All log data greater than this SCN has been read and saved
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_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 lists the tables that will be skipped by log apply services.

Column	Datatype	NULL	Description
ERROR	VARCHAR2(1)		Indicates whether the statement should be skipped (Yes) or just return errors for the statement (No)
STATEMENT_OPT	VARCHAR2(30)		Specifies the type of statement that should be skipped. It must be one of the AUDIT_OPTIONS.
OWNER	VARCHAR2(30)		Name of the schema under which this skip option should be used
NAME	VARCHAR2(30)		Name of the option under which this skip option should be used
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 lists the skip settings chosen.

Column	Datatype	NULL	Description
XIDUSN	NUMBER		Transaction ID undo segment number
XIDSLT	NUMBER		Transaction ID slot number
XIDSQN	NUMBER		Transaction ID sequence number

DBA_LOGSTDBY_UNSUPPORTED

DBA_LOGSTDBY_UNSUPPORTED identifies 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.

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)		Data type of the unsupported column

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 2-66

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 2-67

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 2-68.

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 2-69.

> **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 Oracle8*i* and that were never altered to enable query rewrite.

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 2-71.

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 2-71.

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 2-72.

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

Column	Datatype	NULL	Description
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 2-73

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 2-74

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 2-75

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 2-79.

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
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	VARCHAR2(3)		Auditing REFERENCE WHENEVER SUCCESSFUL/UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL/UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing READ 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 2-79

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" on page 2-80.

DBA_OBJECTS

DBA_OBJECTS describes all objects in the database. Its columns are the same as those in "ALL_OBJECTS" on page 2-82.

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 2-83.

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 2-84.

DBA_OPBINDINGS

DBA_OPBINDINGS describes bindings of all operators in the database. Its columns are the same as those in "ALL_OPBINDINGS" on page 2-84.

DBA_OPERATOR_COMMENTS

DBA_OPERATOR_COMMENTS lists all comments for user-defined operators in the database. Its columns are the same as those in ALL_OPERATOR_COMMENTS.

See Also: "ALL_OPERATOR_COMMENTS" on page 2-85

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 2-85

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
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

Column	Datatype	NULL	Description
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 the all the outlines in the database. Its columns are the same as those in "ALL_OUTLINE_HINTS" on page 2-86.

DBA_OUTLINES

DBA_OUTLINES describes all outlines in the database. Its columns are the same as those in "ALL_OUTLINES" on page 2-86.

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 2-87.

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 2-88.

DBA_PART_INDEXES

DBA_PART_INDEXES provides object-level partitioning information for all partitioned indexes in the database. Its columns are the same as those in "ALL_PART_INDEXES" on page 2-89.

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" on page 2-91.

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 2-91.

DBA_PART_TABLES

DBA_PART_TABLES provides object-level partitioning information for all partitioned tables in the database. Its columns are the same as those in "ALL_PART_TABLES" on page 2-92.

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 2-94.

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 2-94

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

Column	Datatype	NULL	Description
BRANCHID	RAW(64)		The branch qualifier (bqual) of the transaction identifier

DBA_POLICIES

DBA_POLICIES describes all security policies in the database. Its columns are the same as those in "ALL_POLICIES" on page 2-95.

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 2-96.

DBA_PROFILES

DBA_PROFILES displays all profiles and their limits.

Column	Datatype	NULL	Description
PROFILE	VARCHAR2(30)	NOT NULL	Profile name

Column	Datatype	NULL	Description
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 propagation jobs in the database. Its columns are the same as those in ALL_PROPAGATION.

See Also: "ALL_PROPAGATION" on page 2-97

DBA_PROXIES

DBA_PROXIES displays Information about all proxy connections in the system.

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
CREDENTIAL	VARCHAR2(18)		Credential passed by the proxy for the client (if any)
TYPE	VARCHAR2(5)		Type of the credential
VERSION	VARCHAR2(1)		Version of the credential
AUTHORIZATION_CONSTRAINT	VARCHAR2(35)		Indicates the proxy's authority to exercise roles on client's behalf.
ROLE	VARCHAR2(30)		Name of the role referenced in authorization constraint

See Also: "USER_PROXIES" on page 2-260

DBA_PUBLISHED_COLUMNS

DBA_PUBLISHED_COLUMNS describes all existing source column tables. You must have the SELECT_CATALOG_ROLE privilege to access this view.

Column	Datatype	Description
CHANGE_SET_NAME	VARCHAR2(30)	Name of the change set
CHANGE_TABLE_ SCHEMA	VARCHAR2(30)	Associated change table owner
CHANGE_TABLE_NAME	VARCHAR2(31)	Associated change table name
PUB_ID	NUMBER	Publication identifier (change table object number)
SOURCE_SCHEMA_NAME	VARCHAR2(30)	Table owner in the source system
SOURCE_TABLE_NAME	VARCHAR2(31)	Table name in the source system
COLUMN_NAME	VARCHAR2(30)	Column name
DATA_TYPE	VARCHAR2(30)	Column datatype
DATA_LENGTH	NUMBER	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	CHAR(1)	Nulls allowed (Y | N)?

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

Column	Datatype	NULL	Description
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
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

Column	Datatype	NULL	Description
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 queues in all queue tables in the database. Its columns are the same as those in "ALL_QUEUE_TABLES" on page 2-98.

> **See Also:** *Oracle9i Application Developer's Guide - Advanced Queuing* for more information about this view and Advanced Queuing

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 2-99.

> **See Also:** *Oracle9i Application Developer's Guide - Advanced Queuing* 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_REFRESH

DBA_REFRESH describes all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH" on page 2-100.

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 2-101.

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 2-102.

DBA_REGISTERED_MVIEW_GROUPS

DBA_REGISTERED_MVIEW_GROUPS lists all the materialized view repgroups at this site.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the materialized view replication group
MVIEW_SITE	VARCHAR2(128)		Site of the master of the materialized view repgroup
GROUP_COMMENT	VARCHAR2(80)		Description of the materialized view repgroup
VERSION	VARCHAR2(8)		Version of the materialized view repgroup
FNAME	VARCHAR2(30)		Name of the flavor of the materialized view object group
OWNER	VARCHAR2(30)		Owner of the materialized view 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 2-103

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: ■ 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
PARENT_ID	VARCHAR2(30)		Parent component identifier

See Also: "USER_REGISTRY" on page 2-261

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
COMP_ID	VARCHAR2(4000)		Component identifier
VERSION	VARCHAR2(30)		Component version loaded
STATUS	VARCHAR2(11)		Component status: ■ 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

Column	Datatype	NULL	Description
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
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 on which this row was insert into the repair table
FIX_TIMESTAMP	DATE	NULL	Date on which 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 Resumable Statement
INSTANCE_ID	NUMBER		Instance Number of Resumable Statement
COORD_SESSION_ID	NUMBER		Session Identifier of Parallel Coordinator
COORD_INSTANCE_ID	NUMBER		Instance Number on which the Parallel Coordinator is Running
SQL_TEXT	VARCHAR2(1000)		The resumable statement, selected from the V$SQL view.

Column	Datatype	NULL	Description
NAME	VARCHAR2(4000)		The name given in the resumable clause of this resumable statement.
STATUS	VARCHAR2(16)		The status of the resumable statement. Its value can be one of RUNNING, SUSPENDED, ABORTED, ABORTING, TIMEOUT
ERROR_NUMBER	NUMBER		The error code of the last correctable error. When STATUS is equal to RUNNING, its value will be set to 0.
ERROR_MSG	VARCHAR2(4000)		The error message corresponding to ERROR_NUMBER. It will be set to NULL when ERROR_NUMBER is NULL.
ERROR_PARAMETER1	VARCHAR2(80)		The first parameter for the error message. NULL if no error.
ERROR_PARAMETER2	VARCHAR2(80)		The second parameter for the error message. NULL if no error.
ERROR_PARAMETER3	VARCHAR2(80)		The third parameter for the error message. NULL if no error.
ERROR_PARAMETER4	VARCHAR2(80)		The forth parameter for the error message. NULL if no error.
ERROR_PARAMETER5	VARCHAR2(80)		The fifth parameter for the error message. NULL if no error.
START_TIME	DATE		The start local time of the resumable statement.
SUSPEND_TIME	DATE		The last local time when the resumable statement was suspended. It is initialized to NULL.
RESUME_TIME	DATE		The last local time when the suspended resumable statement is resumed. It is initialized to NULL.
TIMEOUT	NUMBER		Timeout of the resumable statement

See Also: "USER_RESUMABLE" on page 2-262

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

Column	Datatype	NULL	Description
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 lists roles granted to users and roles.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)		Grantee name, user or role receiving the grant
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Granted role name
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option (YES | NO)
DEFAULT_ROLE	VARCHAR2(3)		Whether the role is designated as a DEFAULT ROLE for the user (YES | NO)

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	ID 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	NOT NULL	Percent increase for extent size
STATUS	VARCHAR2(16)		Rollback segment status
INSTANCE_NUM	VARCHAR2(40)		Rollback segment owning Oracle Real Application Cluster instance number
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the segment header

DBA_RSRC_CONSUMER_GROUP_PRIVS

DBA_RSRC_CONSUMER_GROUP_PRIVS lists 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 lists all 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)		(YES/NO) Indicates whether grant was with the GRANT option
INITIAL_GROUP	VARCHAR2(3)		(YES/NO) Indicates whether consumer group is designated as the default for this user or role

See Also: *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_CONSUMER_GROUPS

DBA_RSRC_CONSUMER_GROUPS lists 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: PENDING if the group is part of the pending area, ACTIVE otherwise.
MANDATORY	VARCHAR2(3)		Whether the consumer group is mandatory (YES | NO)

DBA_RSRC_MANAGER_SYSTEM_PRIVS

DBA_RSRC_MANAGER_SYSTEM_PRIVS lists 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 SQL GRANT statement.)

Related View

- USER_RSRC_MANAGER_SYSTEM_PRIVS lists users who are granted system privilege 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
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option

See Also: *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_PLAN_DIRECTIVES

DBA_RSRC_PLAN_DIRECTIVES lists all resource plan directives in the database.

See Also:

- *Oracle9i Database Administrator's Guide* for information on resource plans in general
- *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

Column	Datatype	Description
PLAN	VARCHAR2(30)	Name of the plan to which this directive belongs
GROUP_OR_SUBPLAN	VARCHAR2(30)	Name of the consumer group or subplan referred to
TYPE	VARCHAR2(14)	Whether GROUP_OR_SUBPLAN refers to a consumer group or 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
QUEUING_P1	NUMBER	First parameter for the queuing resource allocation method

Column	Datatype	Description
PARALLEL_DEGREE_LIMIT_P1	NUMBER	First parameter for the parallel degree limit resource allocation method
SWITCH_GROUP	VARCHAR2(30)	Group to switch into
SWITCH_TIME	NUMBER	Amount of run time before session is automatically switched
SWITCH_ESTIMATE	VARCHAR2(5)	(TRUE/FALSE) TRUE indicates estimated execution time should be used for switch criteria
MAX_EST_EXEC_TIME	NUMBER	First parameter for the maximum estimated execution time
UNDO_POOL	NUMBER	Undo pool size for the consumer group
COMMENTS	VARCHAR2(2000)	Text comment on the plan directive
STATUS	VARCHAR2(30)	Status of the directive: PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)	Whether the plan is mandatory (YES | NO)

DBA_RSRC_PLANS

DBA_RSRC_PLANS lists all resource plans in the database. For a list of currently active plans, see "V$RSRC_PLAN" on page 3-120.

See Also:

- *Oracle9i Database Administrator's Guide* for information on resource plans in general
- *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

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

Column	Datatype	NULL	Description
STATUS	VARCHAR2(30)		Status of the plan: PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)		(YES/NO) Indicates whether the plan is mandatory

DBA_RULE_SET_RULES

DBA_RULE_SET_RULES describes all 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 2-104

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 2-105

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 2-106

DBA_SEGMENTS

DBA_SEGMENTS describes storage allocated for all database segments.

Related View

- USER_SEGMENTS describes storage allocated for database 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

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(17)		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
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

DBA_SEQUENCES

DBA_SEQUENCES describes all sequences in the database. Its columns are the same as those in "ALL_SEQUENCES" on page 2-106.

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 2-107.

DBA_SOURCE_TABLES

DBA_SOURCE_TABLES enables publishers to see all existing (published) source tables. You must have the SELECT_CATALOG_ROLE privilege to view this table.Its columns are the same as those in ALL_SOURCE_TABLES on page 2-108.

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 2-108

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 2-109

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 2-110

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

Column	Datatype	NULL	Description
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 2-111.

DBA_STREAMS_GLOBAL_RULES

DBA_STREAMS_GLOBAL_RULES displays information about all global rules created by Streams capture/apply/propagation processes. Its columns are the same as those in ALL_STREAMS_GLOBAL_RULES.

See Also: "ALL_STREAMS_GLOBAL_RULES" on page 2-112

DBA_STREAMS_SCHEMA_RULES

DBA_STREAMS_SCHEMA_RULES displays information about all schema rules created by Streams capture/apply/propagation processes. 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 rules created by Streams processes and jobs on all tables in the database. Its columns are the same as those in ALL_STREAMS_TABLE_RULES.

See Also: "ALL_STREAMS_TABLE_RULES" on page 2-113

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 2-111.

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 2-115.

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" on page 2-116.

DBA_SUBSCRIBED_COLUMNS

DBA_SUBSCRIBED_COLUMNS enables publishers to view the columns of published tables to which they have subscribed. You must have the SELECT_CATALOG_ROLE privilege to access this view. The columns in DBA_SUBSCRIBED_COLUMNS are the same as those in "ALL_SUBSCRIBED_COLUMNS" on page 2-116.

DBA_SUBSCRIBED_TABLES

DBA_SUBSCRIBED_TABLES enables publishers to view all of the published tables to which they have subscribed. Its columns are the same as those in "ALL_SUBSCRIBED_TABLES" on page 2-117.

DBA_SUBSCRIPTIONS

DBA_SUBSCRIPTIONS enables publishers to see all subscriptions. You must have the SELECT_CATALOG_ROLE privilege to access this view. Except for the USERNAME column, which identifies the user name of the subscriber, DBA_SUBSCRIPTIONS has the same columns as those in "ALL_SUBSCRIPTIONS" on page 2-117.

DBA_SYNONYMS

DBA_SYNONYMS describes all synonyms in the database. Its columns are the same as those in "ALL_SYNONYMS" on page 2-119.

DBA_SYS_PRIVS

DBA_SYS_PRIVS lists system privileges granted to users and roles. This view does not display the USERNAME column.

Related View

- USER_SYS_PRIVS lists 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 2-207. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 2-119.

DBA_TAB_COLUMNS

DBA_TAB_COLUMNS describes columns of all tables, views, and clusters in the database. Its columns are the same as those in "ALL_TAB_COLUMNS" on page 2-120. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_TAB_COMMENTS

DBA_TAB_COMMENTS describes comments on all tables and views in the database. Its columns are the same as those in "ALL_TAB_COMMENTS" on page 2-122.

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 2-122.

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 2-123.

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 2-124.

DBA_TAB_PRIVS

DBA_TAB_PRIVS lists all grants on all objects in the database. Its columns are the same as those in "ALL_TAB_PRIVS" on page 2-126.

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 2-127.

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 SQL ANALYZE statement.

See Also: "ALL_TABLES" on page 2-129

DBA_TABLESPACES

DBA_TABLESPACES describes all tablespaces in the database.

Related View

- USER_TABLESPACES describes all 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	Tablespace name
BLOCK_SIZE	NUMBER	NOT NULL	Tablespace block size
INITIAL_EXTENT	NUMBER		Default initial extent size
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 the tablespace
STATUS	VARCHAR2(9)		Tablespace status: ■ ONLINE ■ OFFLINE ■ READ ONLY
CONTENTS	VARCHAR2(9)		Tablespace contents: ■ UNDO ■ PERMANENT ■ TEMPORARY
LOGGING	VARCHAR2(9)		Default logging attribute
FORCE_LOGGING	VARCHAR2(3)		Indicates whether the tablespace is under force logging mode (YES) or not (NO)

Column	Datatype	NULL	Description
EXTENT_MANAGEMENT	VARCHAR2(10)		Extent management tracking: ■ DICTIONARY ■ LOCAL
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for this tablespace
PLUGGED_IN	VARCHAR2(3)		Indicates whether the tablespace is plugged in (YES) or not (NO)
SEGMENT_SPACE_MANAGEMENT	VARCHAR2(6)		Segment space management tracking: ■ MANUAL - Segment space in the tablespace is managed using free lists ■ AUTO - Segment space in the tablespace is managed using bitmaps

See Also: "USER_TABLESPACES" on page 2-268

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		ID 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: AVAILABLE
RELATIVE_FNO	NUMBER		Tablespace-relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator (YES | NO)
MAXBYTES	NUMBER		maximum size of the file in bytes
MAXBLOCKS	NUMBER		Maximum size of the file in Oracle database blocks
INCREMENT_BY	NUMBER		Default increment for autoextension
USER_BYTES	NUMBER		Size of the useful portion of file in bytes
USER_BLOCKS	NUMBER		Size of the useful portion of file in Oracle database blocks

DBA_TRANSFORMATIONS

DBA_TRANSFORMATIONS provides information about all message transformations in the database. These transformations are specified with Advanced Queuing operations like enqueue, dequeue and subscribe to automatically integrate transformations in AQ messaging. DBA_TRANSFORMATIONS is available only to users with DBA privileges.

Related View

- USER_TRANSFORMATIONS lists information about transformations owned by particular users. It does not display the OWNER column.

Column	Datatype	NULL	Description
TRANSFORMATION_ID	NUMBER		Unique transformation identifier
OWNER			User owning the transformation
NAME	VARCHAR2(30)	NOT NULL	Name
NAMESPACE	INTEGER		1 for transformations created by Oracle's transformation engine. Transformations from third party transformation engines are in different namespaces.
FROM_TYPE_SCHEMA	VARCHAR2(30)		User owning the source type
FROM_TYPE_NAME	VARCHAR2(30)		Transformation source type
TO_TYPE_SCHEMA	VARCHAR2(30)		User owning the destination type
TO_TYPE_NAME	VARCHAR2(30)		Transformation destination type. The transformation takes an object of the source type and returns an object of the destination type.
TRANSFORMATION_TYPE	TYPE		Type of transformation: SQL, XSL
ATTRIBUTE_NAME	VARCHAR2(30)		Attribute name of the destination type for which the transformation is being specified
TRANSFORMATION_ EXPRESSION	VARCHAR2(4000)		Transformation expression. Can be a SQL expression, PLSQL function or XSL document.
COMMENT	VARCHAR2(4000)		User specified comment

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 2-131.

DBA_TRIGGERS

DBA_TRIGGERS describes all triggers in the database. Its columns are the same as those in "ALL_TRIGGERS" on page 2-132.

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_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 2-133

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 2-134

DBA_TYPE_VERSIONS

DBA_TYPE_VERSIONS describes all 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 2-135

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 2-136

DBA_UNDO_EXTENTS

DBA_UNDO_EXTENTS lists the commit time for each extent in the undo tablespaces.

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	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: ■ 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 2-137.

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 2-137.

See Also: *Oracle9i Database Concepts* for information on updatable join views

DBA_USERS

DBA_USERS describes all users of the database. This view does not display the PROFILE column.

Related View

- USER_USERS describes the current user. This view does not display the PASSWORD column.

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(30)	NOT NULL	Indicates if the account is locked, expired, or unlocked
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	Default tablespace for temporary table
CREATED	DATE	NOT NULL	User creation date
PROFILE	VARCHAR2(30)	NOT NULL	User resource profile name
INITIAL_RSRC_CONSUMER_GROUP	VARCHAR2(30)		The initial resource consumer group for the user

Column	Datatype	NULL	Description
EXTERNAL_NAME	VARCHAR2(4000)		User external name

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 2-138.

DBA_VARRAYS

DBA_VARRAYS describes all the varrays in the database. Its columns are the same as those in "ALL_VARRAYS" on page 2-139.

DBA_VIEWS

DBA_VIEWS displays the text of all views in the database. Its columns are the same as those in "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

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
SEQ#	NUMBER		Sequence number in the dependency tree. Used for ordering queries. **See Also:** "IDEPTREE" on page 2-226

DICT

DICT is a synonym for DICTIONARY.

See Also: "DICTIONARY" on page 2-217

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 2-183 and "USER_OBJECT_SIZE" on page 2-256.

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

FILE_LOCK

This Oracle Real Application Cluster view displays the mapping of PCM locks to datafiles as specified in initialization parameter GC_FILES_TO_LOCKS.

See Also: "GC_FILES_TO_LOCKS" on page 1-60 and *Oracle9i Real Application Clusters Concepts*

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FILE_NAME	VARCHAR2(513)		The datafile name
TS_NAME	VARCHAR2(30)	NOT NULL	The tablespace name for the datafile
START_LK	NUMBER		The first lock corresponding to the datafile
NLOCKS	NUMBER		The number of PCM locks allocated to the datafile
BLOCKING	NUMBER		The number of contiguous blocks protected by each PCM lock on the datafile

FILE_PING

This Oracle Real Application Cluster view displays the number of blocks pinged per datafile. You can use this information to determine access usage of existing datafiles for better settings of GC_FILES_TO_LOCKS.

See Also: "GC_FILES_TO_LOCKS" on page 1-60 and *Oracle9i Real Application Clusters Concepts*

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FILE_NAME	NUMBER		The file name
TS_NAME	VARCHAR2(30)	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FREQUENCY	NUMBER		The ping count
X_2_NULL	NUMBER		Number of lock conversions from Exclusive-to-NULL for all blocks in the file
X_2_NULL_FORCED_WRITE	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions

Column	Datatype	NULL	Description
X_2_NULL_FORCED_STALE	NUMBER		Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER		Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITES	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
X_2_SSX	NUMBER		Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks in the file
X_2_SSX_FORCED_WRITES	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Sub Shared Exclusive conversions
S_2_NULL	NUMBER		Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED_STALE	NUMBER		Number of times a block in the file was made STALE due to Shared-to-NULL conversions
SS_2_NULL	NUMBER		Number of lock conversions from Sub Shared-to-NULL for all blocks in the file
WRB	NUMBER		Number of times the instance received a write single buffer cross instance call for this file
WRB_FORCED_WRITE	NUMBER		Number of blocks written due to write single buffer cross instance calls for this file
RBR	NUMBER		Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER		Number of blocks written due to resuse block range cross instance calls for this file
RBR_FORCED_STALE	NUMBER		Number of times a block in this file was made STALE due to resuse block range cross instance calls
CBR	NUMBER		Number of times the instance received a checkpoint block range cross instance call for this file
CBR_FORCED_WRITE	NUMBER		Number of blocks in this file which were written due to checkpoint cross range cross instance calls
NULL_2_X	NUMBER		Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER		Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file
SSX_2_X	NUMBER		Number of lock conversions from Sub Shared Exclusive-to-Exclusive for all blocks of the specified file
N_2_S	NUMBER		Number of lock conversions from NULL-to-Shared for all blocks of the specified file

Column	Datatype	NULL	Description
N_2_SS	NUMBER		Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file

FILEXT$

FILEXT$ is the equivalent of DBA_DATA_FILES. Oracle Corporation recommends you use DBA_DATA_FILES instead of FILEXT$.

See Also: "DBA_DATA_FILES"

Column	Datatype	NULL	Description
FILE#	NUMBER	NOT NULL	Absolute file number
MAXEXTEND	NUMBER		Maximum file size
INC	NUMBER		Increment amount

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

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.

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

Column	Datatype	NULL	Description
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
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
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 2-249

INDEX_HISTOGRAM

INDEX_HISTOGRAM contains information from the ANALYZE INDEX ... VALIDATE STRUCTURE statement.

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.

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, a 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

Column	Datatype	NULL	Description
LF_ROWS_LEN	NUMBER		Sum of the lengths of all the leaf rows
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

NLS_DATABASE_PARAMETERS

NLS_DATABASE_PARAMETERS lists permanent NLS parameters of the database.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)	NOT NULL	Parameter name

Column	Datatype	NULL	Description
VALUE	VARCHAR2(30)		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(30)		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(30)		Parameter value

OBJ

OBJ is a synonym for USER_OBJECTS.

See Also: "USER_OBJECTS" on page 2-256

PARSED_PIECES

Oracle accesses this view to create views about object size.

See Also: "DBA_OBJECT_SIZE" on page 2-183 and "USER_OBJECT_SIZE" on page 2-256

PARSED_SIZE

Oracle accesses this view to create views about object size.

See Also: "DBA_OBJECT_SIZE" on page 2-183 and "USER_OBJECT_SIZE" on page 2-256

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 that the EXPLAIN PLAN statement was issued
REMARKS	VARCHAR2(80)		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; 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)		The 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)		The cost of the current operation estimated by the cost-based optimizer (CBO)
CARDINALITY	NUMBER(38)		The number of rows returned by the current operation (estimated by the CBO)

Column	Datatype	NULL	Description
BYTES	NUMBER(38)		The number of bytes returned by the current operation
OTHER_TAG	VARCHAR2(255)		OTHER_TAG describes the function of the SQL text in the OTHER column. Values for OTHER_TAG are: ■ SERIAL - the SQL is the text of a locally-executed, serial query plan. Currently, SQL is not loaded in OTHER for this case. ■ SERIAL_FROM_REMOTE - the SQL text shown in the OTHER column will be executed at a remote site. ■ PARALLEL_COMBINED_WITH_PARENT - the parent of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_COMBINED_WITH_CHILD - the child of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_TO_SERIAL - the SQL text shown in the OTHER column is the top-level of the parallel plan. ■ PARALLEL_TO_PARALLEL - the SQL text shown in the OTHER column is executed and output in parallel. ■ PARALLEL_FROM_SERIAL - this operation consumes data from a serial operation and outputs it in parallel.
PARTITION_START	VARCHAR2(255)		The start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(255)		The stop partition of a range of accessed partitions
PARTITION_ID	NUMBER(38)		The 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)		The distribution method
CPU_COST	NUMBER		User-defined CPU cost
IO_COST	NUMBER		User-defined I/O cost

PLUGGABLE_SET_CHECK

PLUGGABLE_SET_CHECK contains pluggable set checks.

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
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
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

Column	Datatype	NULL	Description
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 an 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
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

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 and maps the resource name to the resource number.

Column	Datatype	NULL	Description
RESOURCE#	NUMBER	NOT NULL	Numeric resource code

Column	Datatype	NULL	Description
TYPE#	NUMBER	NOT NULL	Name of type
NAME	VARCHAR2(32)	NOT NULL	Name of resource

ROLE_ROLE_PRIVS

ROLE_ROLE_PRIVS describes 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
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 contains information about 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 contains information about 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

Column	Datatype	NULL	Description
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 2-263

SESSION_CONTEXT

SESSION_CONTEXT lists attributes and the values set for the current session.

See Also: *Oracle9i 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 2-183 and "USER_OBJECT_SIZE" on page 2-256

STMT_AUDIT_OPTION_MAP

STMT_AUDIT_OPTION_MAP lists auditing option type codes.

Column	Datatype	NULL	Description
OPTION#	NUMBER	NOT NULL	Numeric auditing option type code
NAME	VARCHAR2(40)	NOT NULL	Name of the auditing option
PROPERTY	NUMBER	NOT NULL	

SYN

SYN is a synonym for USER_SYNONYMS.

SYNONYMS is included for compatibility with Oracle version 5. Oracle Corporation recommends that you do not use this view.

SYSCATALOG

SYSCATALOG is included for compatibility with Oracle version 5. Oracle Corporation recommends that you do not use this view.

SYSFILES

SYSFILES is included for compatibility with Oracle version 5. Oracle Corporation recommends that you do not use this view.

SYSSEGOBJ

SYSSEGOBJ is included for compatibility with Oracle version 5. Oracle Corporation recommends that you do not use this view.

SYSTEM_PRIVILEGE_MAP

SYSTEM_PRIVILEGE_MAP contains information about system privilege codes.

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	

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 with Oracle version 5. Oracle Corporation recommends that you do not use this view.

TABLE_PRIVILEGE_MAP

TABLE_PRIVILEGE_MAP contains information about access privilege codes.

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)

TABLE_PRIVILEGES

TABLE_PRIVILEGES contains information on grants on objects for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee. This view is included for compatibility with Oracle version 6. Oracle Corporation recommends that you do not use this view.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access is 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
SELECT_PRIV	VARCHAR2(1)		Permission to select from the object
INSERT_PRIV	VARCHAR2(1)		Permission to insert into the object
DELETE_PRIV	VARCHAR2(1)		Permission to delete from the object
UPDATE_PRIV	VARCHAR2(1)		Permission to update the object
REFERENCES_PRIV	VARCHAR2(1)		Permission to reference the object
ALTER_PRIV	VARCHAR2(1)		Permission to alter the object
INDEX_PRIV	VARCHAR2(1)		Permission to create or drop an index on the object
CREATED	VARCHAR2(0)		Timestamp for the grant

TABS

TABS is a synonym for USER_TABLES.

See Also: "USER_TABLES" on page 2-268

TABQUOTAS

TABQUOTAS is included for compatibility with Oracle version 5. Oracle Corporation 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 2–3 shows examples of the values returned depending on the status of the servers.

Table 2–3 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: *Oracle9i 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.

See Also: *Oracle9i User-Managed Backup and Recovery Guide*

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

Column	Datatype	NULL	Description
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)
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

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

Column	Datatype	NULL	Description
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

USER_ALL_TABLES

USER_ALL_TABLES contains descriptions of the object tables and relational tables owned by the current user. Its columns are the same as those in "ALL_ALL_TABLES" on page 2-6.

USER_ARGUMENTS

USER_ARGUMENTS lists the arguments in all procedures and functions that are owned by the current user. Its columns are the same as those in "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 2-14.

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 2-149.

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 2-150.

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 2-151.

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 2-153.

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 `DBA_AW_PS`.

See Also:

- "DBA_AW_PS" on page 2-155
- *Oracle9i OLAP User's Guide* for more information about this view

USER_AWS

`USER_AWS` describes the analytic workspaces owned by the current user. Its columns (except for `OWNER`) are the same as those in `DBA_AWS`.

See Also:

- "DBA_AWS" on page 2-155
- *Oracle9i OLAP User's Guide* for more information about this view

USER_BASE_TABLE_MVIEWS

USER_BASE_TABLE_MVIEWS describes all 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 2-15

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 2-18.

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 2-157.

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 2-18.

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 2-19.

USER_COL_COMMENTS

USER_COL_COMMENTS describes comments on columns of tables and views in the current user's schema. Its columns are the same as those in "ALL_COL_COMMENTS" on page 2-20.

USER_COL_PRIVS

USER_COL_PRIVS describes all column object grants for which the current user is the owner, grantor, or grantee. Its columns are the same as those in "ALL_COL_PRIVS" on page 2-21.

USER_COL_PRIVS_MADE

USER_COL_PRIVS_MADE describes all column object grants for which the current user is the grantor. Its columns are the same as those in "ALL_COL_PRIVS_MADE" on page 2-21.

USER_COL_PRIVS_RECD

USER_COL_PRIVS_RECD describes column object grants for which the current user is the grantee. Its columns are the same as those in "ALL_COL_PRIVS_RECD" on page 2-22.

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 2-22.

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 2-23.

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_DB_LINKS

USER_DB_LINKS describes database links owned by the current user. Its columns are the same as those in "ALL_DB_LINKS" on page 2-27.

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 2-29.

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 2-30.

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 2-31.

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 2-31.

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 2-32.

USER_DIMENSIONS

USER_DIMENSIONS describes dimension objects in the user's schema. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-32.

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 2-33.

USER_EVALUATION_CONTEXT_TABLES

USER_EVALUATION_CONTEXT_TABLES describes all tables in all 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 2-34

USER_EVALUATION_CONTEXT_VARS

USER_EVALUATION_CONTEXT_VARS describes all variables in all 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 2-34

USER_EVALUATION_CONTEXTS

USER_EVALUATION_CONTEXTS describes all 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 2-35

USER_EXTENTS

USER_EXTENTS describes extents of the segments belonging to objects owned by the current user. Its columns are a subset of those in "DBA_EXTENTS" on page 2-164.

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 2-36

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 2-36

USER_FREE_SPACE

USER_FREE_SPACE lists the free extents in tablespaces accessible to the user. Its columns are a subset of those in "DBA_FREE_SPACE" on page 2-165.

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 2-37.

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 2-38.

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 2-39.

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 2-41.

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 OWNER) are the same as those in "ALL_INDEXES" on page 2-43.

USER_INDEXTYPE_COMMENTS

USER_INDEXTYPE_COMMENTS lists all comments for 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 2-46

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 2-46

USER_INDEXTYPES

USER_INDEXTYPES describes all indextypes owned by the current user. Its columns are the same as those in ALL_INDEXTYPES.

See Also: "ALL_INDEXTYPES" on page 2-47

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 2-48.

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 2-48

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 2-49

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 2-50

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 2-51

USER_JAVA_IMPLEMENTS

USER_JAVA_IMPLEMENTS 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_IMPLEMENTS.

See Also: "ALL_JAVA_IMPLEMENTS" on page 2-53

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 2-53

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 2-54

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 2-55

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 2-57

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 2-169

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 2-57

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 2-58

USER_JOBS

USER_JOBS describes all jobs owned by the user. Its columns are the same as those in "ALL_JOBS" on page 2-59.

> **See Also:** *Oracle9i 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 2-60.

USER_LIBRARIES

USER_LIBRARIES describes all libraries owned by the current user. Its columns are the same as those in "ALL_LIBRARIES" on page 2-61.

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 2-61.

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 2-63.

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 2-64.

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 2-65.

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 2-66.

USER_METHOD_PARAMS

USER_METHOD_PARAMS describes the method parameters of all 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 2-66

USER_METHOD_RESULTS

USER_METHOD_RESULTS describes the method results of all 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 2-67

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 2-68.

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 2-69.

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 Oracle8*i* and that were never altered to enable query rewrite.

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 2-71.

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 2-71.

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 2-72.

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 2-73

USER_MVIEW_REFRESH_TIMES

USER_MVIEW_REFRESH_TIMES describes refresh times of all 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 2-74

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 2-75

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 2-79.

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 2-181.

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 2-79

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 2-183.

USER_OBJECT_TABLES

USER_OBJECT_TABLES describes the object tables owned by the current user. Its columns are the same as those in "ALL_OBJECT_TABLES" on page 2-80.

USER_OBJECTS

USER_OBJECTS describes all objects owned by the current user. Its columns are the same as those in "ALL_OBJECTS" on page 2-82.

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 2-83.

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 2-84.

USER_OPBINDINGS

USER_OPBINDINGS describes bindings of operators owned by the current user. Its columns are the same as those in "ALL_OPBINDINGS" on page 2-84.

USER_OPERATOR_COMMENTS

USER_OPERATOR_COMMENTS lists all comments for 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 2-85

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 2-85

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 2-86.

USER_OUTLINES

USER_OUTLINES describes all outlines owned by the current user. Its columns are the same as those in "ALL_OUTLINES" on page 2-86.

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 2-87.

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 2-88.

USER_PART_KEY_COLUMNS

USER_PART_KEY_COLUMNS describes the partitioning key columns for partitioned objects owned by the current user. Its columns are the same as those in "ALL_PART_KEY_COLUMNS" on page 2-91.

USER_PART_INDEXES

USER_PART_INDEXES describes object-level partitioning information for all partitioned indexes owned by the current user. Its columns are the same as those in "ALL_PART_INDEXES" on page 2-89.

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 2-91.

USER_PART_TABLES

USER_PART_TABLES provides object-level partitioning information for partitioned tables owned by the current user. Its columns are the same as those in "ALL_PART_TABLES" on page 2-92.

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 2-94.

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 all 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 2-94

USER_POLICIES

USER_POLICIES describes security policies on all objects owned by the current user. Its columns are the same as those in "ALL_POLICIES" on page 2-95.

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 2-96.

USER_PROXIES

USER_PROXIES displays information about connections the current user is allowed to proxy. Its columns are the same as those in DBA_PROXIES.

See Also: "DBA_PROXIES" on page 2-188

USER_PUBLISHED_COLUMNS

USER_PUBLISHED_COLUMNS describes all existing source table columns for which the user has privileges.

Column	Datatype	Description
SOURCE_SCHEMA_NAME	VARCHAR2(30)	Table owner in the source system
SOURCE_TABLE_NAME	VARCHAR2(31)	Table name in the source system
COLUMN_NAME	VARCHAR2(30)	Column name
DATA_TYPE	VARCHAR2(30)	Column datatype
DATA_LENGTH	NUMBER	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	CHAR(1)	Nulls allowed (Y | N)?

USER_QUEUE_SCHEDULES

USER_QUEUE_SCHEDULES lists information about queue schedules. Its columns are the same as those in "DBA_QUEUE_SCHEDULES" on page 2-189.

USER_QUEUE_TABLES

USER_QUEUE_TABLES describes 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" on page 2-98.

See Also: *Oracle9i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing

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 2-99.

> **See Also:** *Oracle9i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing

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 2-100.

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 2-101.

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 2-102.

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 2-103

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 2-193

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 2-195

USER_ROLE_PRIVS

USER_ROLE_PRIVS lists roles granted to the 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)		Granted with ADMIN option (YES | NO)
DEFAULT_ROLE	VARCHAR2(3)		Role is designated as the user's default role (YES | NO)
OS_GRANTED	VARCHAR2(3)		Granted by the operating system (YES | NO; occurs if configuration parameter OS_ROLES = TRUE)

USER_RSRC_CONSUMER_GROUP_PRIVS

USER_RSRC_CONSUMER_GROUP_PRIVS lists all resource consumer groups granted to the user. Its columns are the same as those in "DBA_RSRC_CONSUMER_GROUP_PRIVS" on page 2-198.

USER_RSRC_MANAGER_SYSTEM_PRIVS

USER_RSRC_MANAGER_SYSTEM_PRIVS lists all the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. Its columns are the same as those in "DBA_RSRC_MANAGER_SYSTEM_PRIVS" on page 2-199.

USER_RULE_SET_RULES

USER_RULE_SET_RULES describes all rules in all 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 2-104

USER_RULE_SETS

USER_RULE_SETS describes all 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 2-105

USER_RULES

USER_RULES describes all 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 2-106

USER_SEGMENTS

USER_SEGMENTS lists information about storage allocation for database segments belonging to a user's objects. Its columns are the same as those in "DBA_SEGMENTS" on page 2-202.

USER_SEQUENCES

USER_SEQUENCES describes all sequences owned by the current user. Its columns are the same as those in "ALL_SEQUENCES" on page 2-106.

USER_SOURCE

USER_SOURCE contains text source of all stored objects belonging to the user. Its columns are the same as those in "ALL_SOURCE" on page 2-107.

USER_SOURCE_TABLES

USER_SOURCE_TABLES enables you to see all of the existing source tables to which you have subscribe privileges. Its columns are the same as those in ALL_SOURCE_TABLES on page 2-108.

USER_SQLJ_TYPE_ATTRS

USER_SQLJ_TYPE_ATTRS describes the attributes of all 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 2-108

USER_SQLJ_TYPE_METHODS

USER_SQLJ_TYPE_METHODS describes the methods of all 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 2-109

USER_SQLJ_TYPES

USER_SQLJ_TYPES describes all 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 2-110

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 2-111.

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 2-111.

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 2-115.

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" on page 2-116.

USER_SUBSCRIBED_COLUMNS

USER_SUBSCRIBED_COLUMNS enables publishers to view the columns of published tables to which they have subscribed. Its columns are the same as those in "ALL_SUBSCRIBED_COLUMNS" on page 2-116.

USER_SUBSCRIBED_TABLES

USER_SUBSCRIBED_TABLES enables subscribers to view all published tables to which they have subscribed. Its columns are the same as those in "ALL_SUBSCRIBED_TABLES" on page 2-117.

USER_SUBSCRIPTIONS

USER_SUBSCRIPTIONS enables subscribers to view all of their subscriptions. Its columns are the same as those in "ALL_SUBSCRIPTIONS" on page 2-117.

USER_SYNONYMS

USER_SYNONYMS describes private synonyms (synonyms owned by the current user). Its columns are the same as those in "ALL_SYNONYMS" on page 2-119.

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 2-207.

USER_TAB_COL_STATISTICS

USER_TAB_COL_STATISTICS contains column statistics and histogram information extracted from "USER_TAB_COLUMNS" on page 2-266. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 2-119.

USER_TAB_COLUMNS

USER_TAB_COLUMNS describes the columns of tables, views, and clusters owned by the current user. Its columns are the same as those in "ALL_TAB_COLUMNS" on page 2-120. To gather statistics for this view, use the SQL ANALYZE statement.

USER_TAB_COMMENTS

USER_TAB_COMMENTS describes comments on the tables and views owned by the current user. Its columns are the same as those in "ALL_TAB_COMMENTS" on page 2-122.

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 2-122.

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 2-123.

> **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 2-124.

USER_TAB_PRIVS

USER_TAB_PRIVS describes grants on objects where the current user is the object owner, grantor, or grantee. Its columns are the same as those in "ALL_TAB_PRIVS" on page 2-126.

USER_TAB_PRIVS_MADE

USER_TAB_PRIVS_MADE lists all grants on objects owned by the user. Its columns are the same as those in "ALL_TAB_PRIVS_MADE" on page 2-126.

USER_TAB_PRIVS_RECD

USER_TAB_PRIVS_RECD lists object grants for which the current user is the grantee. Its columns are the same as those in "ALL_TAB_PRIVS_RECD" on page 2-127.

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 2-127.

Note: Statistics are not collected on a per-subpartition basis.

USER_TABLES

USER_TABLES describes all relational tables owned by the current user. Its columns are the same as those in ALL_TABLES. To gather statistics for this view, use the SQL ANALYZE statement.

See Also: "ALL_TABLES" on page 2-129

USER_TABLESPACES

USER_TABLESPACES describes all 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 2-209

USER_TRANSFORMATIONS

USER_TRANSFORMATIONS lists information about transformations owned by particular users. Its columns are the same as those in "DBA_TRANSFORMATIONS" on page 2-211.

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 2-131.

USER_TRIGGERS

USER_TRIGGERS describes all triggers owned by the current. Its columns are the same as those in "ALL_TRIGGERS" on page 2-132.

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 2-212.

USER_TYPE_ATTRS

USER_TYPE_ATTRS describes the attributes of all 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 2-133

USER_TYPE_METHODS

USER_TYPE_METHODS describes the methods of all 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 2-134

USER_TYPE_VERSIONS

USER_TYPE_VERSIONS describes all versions of all 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 2-135

USER_TYPES

USER_TYPES describes all 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 2-136

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 2-137.

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 2-137.

See Also: *Oracle9i Database Concepts* for information on updatable join views

USER_USERS

USER_USERS describes the current user. Its columns are the same as those in "DBA_USERS" on page 2-214.

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 2-138.

USER_VARRAYS

USER_VARRAYS describes the varrays owned by the current user. Its columns are the same as those in "ALL_VARRAYS" on page 2-139.

USER_VIEWS

USER_VIEWS displays the text of views owned by the current user. Its columns are the same as those in "ALL_VIEWS" on page 2-140.

3

Dynamic Performance (V$) Views

This chapter describes the dynamic performance views, which are often referred to as V$ views.

About Dynamic Performance Views

Oracle contains a set of underlying views that are maintained by the 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 username 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 Enterprise Manager and Oracle Trace, 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.

A number of dynamic performance views are relevant only if you are using Oracle Replication. Descriptions of the following Oracle Replication dynamic performance views can be found in the *Oracle9i Replication Management API Reference*.

- V$MVREFRESH
- V$REPLPROP
- V$REPLQUEUE

GV$ Views

For almost every V$ view described in this chapter, Oracle has a corresponding GV$ (global V$) view. In Oracle9*i* Real Application Clusters, querying a GV$ 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:

```
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: *Oracle9i Real Application Clusters Concepts*

The following sections list the columns and public synonyms for the dynamic performance views.

V$ACCESS

This view 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

This view maps instance names to instance numbers for all instances that have the database currently mounted.

Column	Datatype	Description
INST_NUMBER	NUMBER	The instance number
INST_NAME	VARCHAR2(60)	The instance name

V$ACTIVE_SESS_POOL_MTH

This view displays all available active session pool resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the active session pool resource allocation method

V$AQ

This view describes statistics for the queues in the database.

Column	Datatype	Description
QID	NUMBER	The 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'

Column	Datatype	Description
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

This view contains information on redo log files in need of archiving. Each row provides 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 3-78

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
CURRENT	VARCHAR2(3)	This column is obsolete and maintained only for compatibility reasons. 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 describes, for the current instance, all the archived redo log destinations, their current value, mode, and status.

Column	Datatype	Description
DEST_ID	NUMBER	Identifies the log archive destination parameter (1 to 10)
DEST_NAME	VARCHAR2(256)	???

Column	Datatype	Description
STATUS	VARCHAR2(9)	Identifies the current status of the destination: ■ 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
BINDING	VARCHAR2(9)	Specifies how failure will affect the archival operation: ■ MANDATORY - Successful archival is required ■ OPTIONAL - Successful archival is not required (depends on LOG_ARCHIVE_MIN_SUCCEED_DEST)
NAME_SPACE	VARCHAR2(7)	Identifies the scope of parameter setting: ■ SYSTEM - System definition ■ SESSION - Session definition
TARGET	VARCHAR2(7)	Specifies whether the archive destination is local or remote to the primary database: ■ PRIMARY - local ■ STANDBY - remote
ARCHIVER	VARCHAR2(10)	Identifies the archiver process relative to the database where the query is issued: ■ ARC*n* ■ FOREGROUND ■ LGWR ■ RFS
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: ■ ARC*n* ■ FOREGROUND ■ LGWR
REGISTER	VARCHAR2(3)	Indicates whether the archived redo log is registered in the remote destination control file. If the archived redo log is registered, it is available to the managed recovery operation: ■ YES ■ NO
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 re-establish 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: ■ SYNC=PARALLEL ■ SYNC=NOPARALLEL ■ ASYNC
ASYNC_BLOCKS	NUMBER	Number of blocks specified for the ASYNC attribute
AFFIRM	VARCHAR2(3)	Specifies disk I/O mode

Column	Datatype	Description
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.

See Also:

- "LOG_ARCHIVE_DEST" on page 1-75 and "LOG_ARCHIVE_DEST_n" on page 1-76
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-79 and "LOG_ARCHIVE_DEST_STATE_n" on page 1-78
- "STANDBY_ARCHIVE_DEST" on page 1-157
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-82

V$ARCHIVE_DEST_STATUS

V$ARCHIVE_DEST_STATUS displays runtime and configuration information for the archived redo log destinations.

Column	Datatype	Description
DEST_ID	NUMBER	Identifies the log archive destination parameter (1 to 10)
DEST_NAME	VARCHAR2(256)	???
STATUS	VARCHAR2(9)	Identifies the current status of the destination: ■ 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)	Identifies the type of archival destination database: ■ LOCAL - Local to primary database ■ PHYSICAL - Physical standby ■ CROSS-INSTANCE - An instance of the primary

Column	Datatype	Description
DATABASE_MODE	VARCHAR2(15)	Identifies the current mode of the archival destination database: ■ STARTED - Instance started, not mounted ■ MOUNTED - Mounted ■ MOUNTED-STANDBY - Mounted standby ■ OPEN - Open read / write ■ OPEN_READ-ONLY - Open read-only
RECOVERY_MODE	VARCHAR2(7)	Identifies the current mode of media recovery at the archival destination database ■ IDLE - Managed recovery is not active ■ MANUAL - Manual media recovery active ■ MANAGED - Managed recovery is active
PROTECTION_MODE	VARCHAR2(20)	Indicates whether the database is protected: ■ MAXIMUM PROTECTION ■ MAXIMUM AVAILABILITY ■ RESYNCHRONIZATION ■ MAXIMUM PERFORMANCE ■ UNPROTECTED
DESTINATION	VARCHAR2(256)	Specifies the location where the archived redo logs are 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 the use of standby redo logfiles on the standby database (YES) or (NO)

V$ARCHIVE_PROCESSES

This view provides information about the state of the various ARCH processes for the instance.

Column	Datatype	Description
PROCESS	NUMBER	The identifier for the ARCH process for the instance, numbered from 0-9
STATUS	VARCHAR2(10)	The 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

This view 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	The original destination from which the archivelog was generated. 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 this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
FIRST_CHANGE#	NUMBER	First change# in the archived log
FIRST_TIME	DATE	Timestamp of the first change

Column	Datatype	Description
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 this archived log was copied. The online log logical block size is a platform-specific value that is not adjustable by the user.
CREATOR	VARCHAR2(7)	Identifies the creator of the archivelog (ARCH, FRDF, or RMAN)
REGISTRAR	VARCHAR2(7)	Identifies the registrar of the entry (RFS, ARCH, FRGD, RMAN, or SRMN, which is RMAN at standby)
STANDBY_DEST	VARCHAR2(3)	(YES | NO) Indicates if the entry is an archivelog destination
ARCHIVED	VARCHAR2(3)	Indicates that the online redo log was archived (YES) or that RMAN only inspected the log and created a record for future application of redo logs during recovery. **See Also:** *Oracle9i Recovery Manager User's Guide*.
APPLIED	VARCHAR2(3)	(YES | NO) Indicates whether or not the archivelog has been applied to its corresponding standby database. YES indicates it has been applied; NO indicates it has not. 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)	Specifies (YES | NO) 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.
STATUS	VARCHAR2(1)	The status of this archived log. Possible values are: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the archiving completed
DICTIONARY_BEGIN	VARCHAR2(3)	(YES | NO) Indicates whether or not this log contains the start of a LogMiner dictionary. This column also appears in the RC_ARCHIVED_LOG view.

Column	Datatype	Description
DICTIONARY_END	VARCHAR2(3)	(YES | NO) Indicates whether or not this log contains the end of a LogMiner dictionary. This column also appears in the RC_ARCHIVED_LOG view.
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	

V$BACKUP

This view 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

This view 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: *Oracle9i Database Performance Tuning Guide and Reference* 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
DEVICE_TYPE	VARCHAR2(17)	The 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)	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
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 100ths of seconds, 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
READY	NUMBER	The number of asynchronous requests for which a buffer was immediately ready for use
SHORT_WAITS	NUMBER	The number of times that a buffer was not immediately available, but a buffer became available after doing a non-blocking poll for I/O completion
SHORT_WAIT_TIME_TOTAL	NUMBER	The total time, in 100ths of seconds, taken by non-blocking polls for I/O completion
SHORT_WAIT_TIME_MAX	NUMBER	The maximum time taken for a non-blocking poll for I/O completion, in 100ths of seconds
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 100ths of seconds, 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 100ths of seconds

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
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 Oracle (YES) or Oracle had already discovered this corrupt block and marked it as corrupt in the database (NO). Note that when a corrupt block is encountered in a backup, and was not already marked corrupt by Oracle, 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: ■ 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

This view displays backup datafile and backup control file information 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 change of the datafile
CREATION_TIME	DATE	Creation timestamp of the datafile
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# 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
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	The RECID of the oldest offline range record in this backup control file. 0 for datafile backups.
COMPLETION_TIME	DATE	The time completed
CONTROLFILE_TYPE	VARCHAR2(1)	B indicates normal copies S indicates standby copies

V$BACKUP_DEVICE

This view 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(512)	Name of the backup device

V$BACKUP_PIECE

This view displays information about backup pieces from the control file. Each backup set consist 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(81)	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

Column	Datatype	Description
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	The starting time
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The 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.

V$BACKUP_REDOLOG

This view 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
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

Column	Datatype	Description
BLOCK_SIZE	NUMBER	Size of the log blocks in bytes

V$BACKUP_SET

V$BACKUP_SET displays backup set information 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 archivelog backups.
PIECES	NUMBER	Number of distinct backup pieces in the backup set
START_TIME	DATE	The starting time
COMPLETION_TIME	DATE	The 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.

Column	Datatype	Description
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

V$BACKUP_SPFILE

V$BACKUP_SPFILE displays information about server parameter files in backup sets.

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

This view 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: *Oracle9i Database Performance Tuning Guide and Reference* 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
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.
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 100ths of seconds, 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 100ths of seconds, 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

This view describes the background processes.

Column	Datatype	Description
PADDR	RAW(4)	Address of the process state object
NAME	VARCHAR2(5)	Name of this background process
DESCRIPTION	VARCHAR2(64)	Description of the background process
ERROR	NUMBER	Error encountered

V$BH

This is an Oracle9*i* 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 filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	The class number
STATUS	VARCHAR2(1)	Status of the buffer: ■ 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 but is retained for historical compatibility.
LOCK_ELEMENT_ADDR	RAW(4)	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_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 made re-read from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on this block in lock mode.

Column	Datatype	Description
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had dirtied the block and another instance had requested the PCM lock on the block in 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 but is retained for historical compatibility.
OBJD	NUMBER	Database object number of the block that the buffer represents
TS#	NUMBER	Tablespace number of block

See Also: *Oracle9i Real Application Clusters Concepts*

V$BUFFER_POOL

This view displays information about all buffer pools available for the instance. The "sets" pertain to the number of LRU latch sets.

See Also: "DB_BLOCK_SIZE" on page 1-36

Column	Datatype	Description
ID	NUMBER	Buffer pool ID number
NAME	VARCHAR2(20)	Buffer pool name. Possible values: DEFAULT, KEEP, RECYCLE. Note: Currently, KEEP and RECYCLE pools only exist for the standard block size. All non-standard block size pools are DEFAULT.
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 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

Column	Datatype	Description
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

V$BUFFER_POOL_STATISTICS

V$BUFFER_POOL_STATISTICS displays information about all buffer pools available for the instance. The "sets" pertain to the number of LRU latch sets.

See Also: "DB_CACHE_SIZE" on page 1-38

Column	Datatype	Description
ID	NUMBER	Buffer pool ID number
NAME	VARCHAR2(20)	Buffer pool name
SET_MSIZE	NUMBER	Buffer pool maximum set size
CNUM_REPL	NUMBER	Number of buffers on replacement list
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

Column	Datatype	Description
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

V$CACHE

This is an Oracle9*i* 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.

See Also: *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(1)	Status of block: ■ 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 but is retained for historical compatibility.

Column	Datatype	Description
FORCED_READS	NUMBER	The forced reads
FORCED_WRITES	NUMBER	The forced writes
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	The name of the partition; NULL for non-partitioned objects
KIND	VARCHAR2(12)	Type of database object: ■ 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)	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_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.

V$CACHE_LOCK

This is an Oracle9*i* 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.

See Also:

- *Oracle9i Real Application Clusters Concepts*
- "V$CACHE" on page 3-24

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
STATUS	VARCHAR2(4)	Status of block: ■ 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: ■ 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)	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_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.
FORCED_READS	NUMBER	Number of times the block had to be made re-read from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on this block in lock mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had dirtied the block and another instance had requested the PCM lock on the block in conflicting mode.

Column	Datatype	Description
INDX	NUMBER	Platform-specific lock manager identifier
CLASS	NUMBER	Platform-specific lock manager identifier

V$CACHE_TRANSFER

This is an Oracle9*i* 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.

See Also: "V$CACHE" on page 3-24 and *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query "DBA_DATA_FILES" on page 2-159 or "V$DBFILE" on page 3-44)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(4)	Status of block: ■ 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 but is retained for historical compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk 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

Column	Datatype	Description
KIND	VARCHAR2(15)	Type of database object **See Also:** Table 3–1 on page 3-76
OWNER#	NUMBER	Owner number
GC_ELEMENT_ADDR	RAW(4)	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.
GC_ELEMENT_NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer

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)	Circuit address
DISPATCHER	RAW(4)	Current dispatcher process address
SERVER	RAW(4)	Current server process address
WAITER	RAW(4)	Address of server process that is waiting for the (currently busy) circuit to become available
SADDR	RAW(4)	Address of session bound to the circuit
STATUS	VARCHAR2(16)	Status of the circuit: ■ 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: ■ 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

Column	Datatype	Description
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(16)	The presentation protocol used by the client and server

V$CLASS_CACHE_TRANSFER

V$CLASS_CACHE_TRANSFER displays the number of blocks pinged per block class. Use this view to compare contentions for blocks in different classes.

Column	Datatype	Description
CLASS	VARCHAR2(18)	The block class
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks of the specified CLASS
X_2_NULL_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-NULL conversions
X_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the CLASS was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks of the specified CLASS
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-Shared conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks of the specified CLASS
S_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the CLASS was made STALE due to Shared-to-NULL conversions
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified CLASS
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified CLASS
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified CLASS

V$COMPATIBILITY

This view displays features in use by the database instance that may prevent downgrading to a previous release. This is the dynamic (SGA) version of this

information, and may not reflect features that other instances have used, and may include temporary incompatibilities (like UNDO segments) that will not exist after the database is shut down cleanly.

Column	Datatype	Description
TYPE_ID	VARCHAR2(8)	Internal feature identifier
RELEASE	VARCHAR2(60)	Release in which that feature appeared
DESCRIPTION	VARCHAR2(64)	Description of the feature

V$COMPATSEG

This view lists the permanent features in use by the database that will prevent moving back to an earlier release.

Column	Datatype	Description
TYPE_ID	VARCHAR2(8)	Internal feature identifier
RELEASE	VARCHAR2(60)	Release in which that feature appeared. The software must be able to interpret data formats added in that release.
UPDATED	VARCHAR2(60)	Release that first used the feature

V$CONTEXT

This view lists set attributes in the current session.

Column	Datatype	Description
NAMESPACE	VARCHAR2(30)	The namespace that the attribute is in
ATTRIBUTE	VARCHAR2(30)	The name of attribute
VALUE	VARCHAR2(4000)	The value of 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(257)	The name of the control file

V$CONTROLFILE_RECORD_SECTION

This view displays information about the control file record sections.

Column	Datatype	Description
TYPE	VARCHAR2(17)	Identifies the type of record section: DATABASE, CKPT PROGRESS, REDO THREAD, REDO LOG, DATAFILE, FILENAME, TABLESPACE, LOG HISTORY, OFFLINE RANGE, ARCHIVED LOG, BACKUP SET, BACKUP PIECE, BACKUP DATAFILE, BACKUP REDOLOG, DATAFILE COPY, BACKUP CORRUPTION, COPY CORRUPTION, DELETED OBJECT, or PROXY COPY
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
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

Column	Datatype	Description
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: ■ 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 block server background processes (BSP*n*) used in cache fusion.

Note: This view contains internal diagnostic information for use by Oracle Support Services. It is subject to change without notice.

See Also: *Oracle9i Real Application Clusters Concepts* for more information on these processes and on cache fusion

Column	Datatype	Description
CR_REQUESTS	NUMBER	Number of requests received for a version of a block at a specific SCN ("CR" refers to "consistent-read")
CURRENT_REQUESTS	NUMBER	Number of requests for the most recent (current) version of a block CR_REQUESTS + CURRENT_REQUESTS = total number of requests handled by the BSP*n* processes
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 BSP*n* processes

Column	Datatype	Description
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
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 a BSP*n* process
LIGHT_WORKS	NUMBER	Number of times the light-work rule was evoked. This rule prevents the BSP back ground process from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the BSP process from completing its response to the CR request.

V$DATABASE

V$DATABASE contains database information from the control file.

Column	Datatype	Description
DBID	NUMBER	Database ID calculated when the database is created and stored in all file headers
NAME	VARCHAR2(9)	Name of the database
CREATED	DATE	Creation date
RESETLOGS_CHANGE#	NUMBER	Change number at open resetlogs
RESETLOGS_TIME	DATE	Timestamp of open resetlogs
PRIOR_RESETLOGS_CHANGE#	NUMBER	Change number at prior resetlogs
PRIOR_RESETLOGS_TIME	DATE	Timestamp of prior resetlogs

Column	Datatype	Description
LOG_MODE	VARCHAR2(12)	Archive log mode (NOARCHIVELOG or ARCHIVELOG)
CHECKPOINT_CHANGE#	NUMBER	Last SCN checkpointed
ARCHIVE_CHANGE#	NUMBER	Last SCN archived
CONTROLFILE_TYPE	VARCHAR2(7)	Type of control file: ■ 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 - the control file changes to this type following a standby database activate or database open after recovery
CONTROLFILE_CREATED	DATE	Control file creation timestamp
CONTROLFILE_SEQUENCE#	NUMBER	Control file sequence number incremented by control file transactions
CONTROLFILE_CHANGE#	NUMBER	Last change number 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
PROTECTION_MODE	VARCHAR2(20)	Protection mode currently in effect for the database: ■ 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)	Aggregated protection mode currently in effect for the database: ■ 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) **Note:**This column is an aggregation of the PROTECTION_MODE of all standby archivelog destinations.
REMOTE_ARCHIVE	VARCHAR2(8)	Value of the REMOTE_ARCHIVE_ENABLE initialization parameter
ACTIVATION#	NUMBER	Number assigned to the database instantiation
DATABASE_ROLE	VARCHAR2(16)	Current role of the database; either primary or standby
ARCHIVELOG_CHANGE#	NUMBER	Highest NEXT_CHANGE# (from the V$ARCHIVED_LOG view) for an archived log
SWITCHOVER_STATUS	VARCHAR2(18)	Indicates whether switchover is allowed: ■ NOT ALLOWED - Either this is a standby database and the primary database has not been switched first or this is a primary database and there are no standby databases. ■ SESSIONS ACTIVE - Indicates that there are active SQL sessions attached to the primary or standby database that need to be disconnected before the switchover operation is permitted. Query the V$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.

Column	Datatype	Description
GUARD_STATUS	VARCHAR2(7)	Protects data from being changed: ■ 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(3)	Ensures that LogMiner will have sufficient information to support chained rows and various storage arrangements such as cluster tables. See *Oracle9i SQL Reference* for additional information about the ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement.
SUPPLEMENTAL_LOG_DATA_PK	VARCHAR2(3)	For all tables with a primary key, ensures that all columns of the primary key are placed into the redo log whenever an update operation is performed. See *Oracle9i SQL Reference* for additional information about the ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement.
SUPPLEMENTAL_LOG_DATA_UI	VARCHAR2(3)	For all tables with a unique key, ensures that if any unique key columns are modified, all other columns belonging to the unique key are also placed into the redo log. See *Oracle9i SQL Reference* for additional information about the ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement.
FORCE_LOGGING	VARCHAR2(3)	Indicates whether the database is under force logging mode (YES) or not (NO)

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.

Column	Datatype	Description
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: ■ 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$DATABASE_INCARNATION displays information about all database incarnations. Oracle creates a new incarnation whenever you open a database with the RESETLOGS option. Records about the current and immediately previous incarnation are also contained in the V$DATABASE view.

Column	Datatype	Description
RESETLOGS_CHANGE#	NUMBER	SCN of the RESETLOGS operation that created this incarnation
RESETLOGS_TIME	DATE	Timestamp of the RESETLOGS operation that created this incarnation
PRIOR_RESETLOGS_CHANGE#	NUMBER	SCN of the previous RESETLOGS operation
PRIOR_RESETLOGS_TIME	DATE	Timestamp of the previous RESETLOGS operation

V$DATAFILE

This view contains datafile information from the control file.

See Also: "V$DATAFILE_HEADER" on page 3-40, which displays information from datafile headers

Column	Datatype	Description
FILE#	NUMBER	File identification number

Column	Datatype	Description
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: ■ 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	Time stamp 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	Time stamp 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 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)	Datafile name
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	The 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)	The 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.

Column	Datatype	Description
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	The 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
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

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.

Column	Datatype	Description
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
NAME	VARCHAR2(512)	Datafile 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. Possible values are: ■ CRASH RECOVERY ■ LTS ■ LAS ■ RMS ■ REMOTE FILE SERVER ■ FETCH ARCHIVE LOG ■ DATA GUARD ■ NETWORK SERVICES
SEVERITY	VARCHAR2(13)	Severity of the event. Possible values are: ■ 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	???
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 1-37

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)
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

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	Owner of the object
NAME	VARCHAR2	Name of the object
DB_LINK	VARCHAR2	Database link name, if any
NAMESPACE	VARCHAR2	Library cache namespace of the object: TABLE/PROCEDURE, BODY, TRIGGER, INDEX, CLUSTER, OBJECT
TYPE	VARCHAR2	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:** "V$SQLAREA" on page 3-155 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 DBMS_SHARED_POOL.KEEP
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.

See Also: "V$DATAFILE" on page 3-37

Column	Datatype	Description
FILE#	NUMBER	File identifier
NAME	VARCHAR2	Name of file

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(13)	Identifies the type of deleted object: ARCHIVED LOG, DATAFILE COPY, BACKUP PIECE, PROXY COPY, BACKUP PIECE AVAILABLE, BACKUP PIECE EXPIRED, PROXY COPY AVAILABLE, or PROXY COPY EXPIRED
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

This view provides information on the dispatcher processes.

Column	Datatype	Description
NAME	VARCHAR2	Name of the dispatcher process
NETWORK	VARCHAR2	Network address of this dispatcher
PADDR	RAW(4)	Process address
STATUS	VARCHAR2	The status of the dispatcher (one of the following): ■ 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	Whether this dispatcher is accepting new connections (YES | NO)
MESSAGES	NUMBER	Number of messages processed by this dispatcher
BYTES	NUMBER	Size in bytes of messages processed by this dispatcher
BREAKS	NUMBER	Number of breaks occurring in this connection
OWNED	NUMBER	Number of circuits owned by this dispatcher
CREATED	NUMBER	Number of circuits created by this dispatcher
IDLE	NUMBER	Total idle time for this dispatcher in hundredths of a second
BUSY	NUMBER	Total busy time for this dispatcher in hundredths of a second
LISTENER	NUMBER	The most recent Oracle error number the dispatcher received from the listener
CONF_INDX	NUMBER	Zero-based index of the DISPATCHERS configuration used by this dispatcher

V$DISPATCHER_RATE

This view provides 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	CHAR	Name of the dispatcher process
PADDR	RAW	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 reconnections / 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

Column	Datatype	Description
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 reconnections/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
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

Column	Datatype	Description
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.
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.

Column	Datatype	Description
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 3-75

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	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: ■ 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: ■ 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_NAME

This view contains information about wait events.

Column	Datatype	Description
EVENT#	NUMBER	The number of the wait event
NAME	VARCHAR2(64)	The name of the wait event
PARAMETER1	VARCHAR2(64)	The description of the first parameter for the wait event
PARAMETER2	VARCHAR2(64)	The description of the second parameter for the wait event
PARAMETER3	VARCHAR2(64)	The description of the third parameter for the wait event

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 an Oracle9*i* 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.

See Also: "GC_FILES_TO_LOCKS" on page 1-60 and *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
STATUS	VARCHAR2(1)	Status of block: ■ 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 but is retained for historical compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk 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	NULL for non-partitioned objects
KIND	VARCHAR2(12)	Type of database object
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4)	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_NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer
LOCK_ELEMENT_CLASS	NUMBER	The lock element class

V$FAST_START_SERVERS

V$FAST_START_SERVERS provides information about all the recovery slaves performing parallel transaction recovery.

See Also: *Oracle9i User-Managed Backup and Recovery Guide*

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

V$FAST_START_TRANSACTIONS

V$FAST_START_TRANSACTIONS contains information about the progress of the transactions that Oracle is recovering.

See Also: *Oracle9i User-Managed Backup and Recovery Guide*

Column	Datatype	Description
USN	NUMBER	The undo segment number of the transaction
SLT	NUMBER	The slot within the rollback segment
SEQ	NUMBER	The incarnation number of the slot
STATE	VARCHAR2(16)	The state of the transaction may be TO BE RECOVERED, RECOVERED, or RECOVERING
UNDOBLOCKSDONE	NUMBER	The number of undo blocks completed on this transaction
UNDOBLOCKSTOTAL	NUMBER	The total number of undo blocks that need recovery
PID	NUMBER	The ID of the current server it has been assigned to
CPUTIME	NUMBER	The time for which recovery has progressed, in seconds
PARENTUSN	NUMBER	The undo segment number of the parent transaction in PDML
PARENTSLT	NUMBER	The slot of the parent transaction in PDML
PARENTSEQ	NUMBER	The sequence number of the parent transaction in PDML

V$FILE_CACHE_TRANSFER

V$FILE_CACHE_TRANSFER displays the number of blocks pinged per datafile. This information in turn can be used to determine access patterns to existing datafiles and deciding new mappings from datafile blocks to PCM locks.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile

Column	Datatype	Description
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks in the file
X_2_NULL_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions
X_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the file was made STALE due to Shared-to-NULL conversions
RBR	NUMBER	Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse block range cross instance calls for this file
RBR_FORCED_STALE	NUMBER	Number of times a block in this file was made STALE due to reuse block range cross instance calls
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file

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

Column	Datatype	Description
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
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
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, 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

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)	The name of the fixed view
VIEW_DEFINITION	VARCHAR2(2000)	The definition of the fixed view

V$GC_ELEMENT

This is an Oracle9*i* Real Application Clusters view. There is one entry in V$GC_ELEMENT for each PCM lock that is used by the buffer cache. The name of the PCM lock that corresponds to a lock element is {'BL', indx, class}.

See Also: *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4)	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 an Oracle9*i* 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: *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4)	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$GCSHVMASTER_INFO

V$GCSHVMASTER_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 Oracle9*i* 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 3-62 for a description of all locks known to the lock manager

Column	Datatype	Description
HANDLE	RAW(4)	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 3–3
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in 100th 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 3–3 ■ 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 100th of a second)

Column	Datatype	Description
CONVERT_COUNT	NUMBER	The number of operations

V$GES_ENQUEUE

This Oracle9*i* Real Application Clusters view describes all locks currently known to lock manager.

See Also: "V$GES_BLOCKING_ENQUEUE" on page 3-59 for a description of all such locks that are currently blocking or being blocked

Column	Datatype	Description
HANDLE	RAW(4)	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 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

Column	Datatype	Description
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 3-69 for statistics about GES latch performance

V$GES_RESOURCE

V$GES_RESOURCE is an Oracle9*i* Real Application Clusters view. It displays information of all resources currently known to the lock manager.

Column	Datatype	Description
RESP	RAW(4)	Resource pointer
RESOURCE_NAME	VARCHAR2(30)	Resource name in hexadecimal for the lock
ON_CONVERT_Q	NUMBER	1 if on convert queue, 0 otherwise
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)	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)	Address of lock state object (raw)
KADDR	RAW(4)	Address of lock (raw)
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

Column	Datatype	Description
STATE	VARCHAR2(18)	State of the branch of the global transaction
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	Net8 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(64)	Name of the parameter
VALUE	VARCHAR2(64)	Value of the parameter
SOURCE	CHAR(1)	Whether the parameter was defined in the agent (A) or server (S)
ENV	CHAR(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	Net8 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$HVMASTER_INFO

V$HVMASTER_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)	The 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 INSTANCE_NUMBER initialization parameter. **See Also:** "INSTANCE_NUMBER"
INSTANCE_NAME	VARCHAR2(16)	Instance name
HOST_NAME	VARCHAR2(64)	Name of the host machine
VERSION	VARCHAR2(17)	RDBMS version
STARTUP_TIME	DATE	Time when instance was started up
STATUS	VARCHAR2(12)	Instance status: ■ 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 MIGRATE
PARALLEL	VARCHAR2(3)	(YES/NO) Indicates whether the instance is mounted in cluster database mode
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
SHUTDOWN_PENDING	VARCHAR2(3)	YES | NO
DATABASE_STATUS	VARCHAR2(17)	The status of the database
INSTANCE_ROLE	VARCHAR2(18)	Describes whether the instance is an active instance (PRIMARY_INSTANCE) or a inactive secondary instance (SECONDARY_INSTANCE), or UNKNOWN if the instance has been started but not mounted

Column	Datatype	Description
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 an Oracle9*i* 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

This view monitors the mechanisms available to users to limit recovery I/O. Those mechanisms are:

- Set the initialization parameter LOG_CHECKPOINT_TIMEOUT
- Set the initialization parameter LOG_CHECKPOINT_INTERVAL
- Set the initialization parameter FAST_START_MTTR_TARGET
- Set the size of the smallest redo log

See Also:

- *Oracle9i Database Performance Tuning Guide and Reference* for more information on limiting recovery I/O, and how MTTR recovery times are determined by Oracle
- LOG_CHECKPOINT_INTERVAL on page 1-86
- FAST_START_MTTR_TARGET on page 1-58

Column	Datatype	Description
RECOVERY_ESTIMATED_IOS	NUMBER	Contains the number of dirty buffers in the buffer cache. In the Standard Edition, this column is always NULL.
ACTUAL_REDO_BLKS	NUMBER	The current actual number of redo blocks required for recovery
TARGET_REDO_BLKS	NUMBER	The 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 only for compatibility reasons. 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), 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	The 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

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)	Address of the latch object
LATCH#	NUMBER	Latch number
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
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
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$LATCH_CHILDREN" on page 3-70
- "V$LATCH_PARENT" on page 3-72

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)	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(64)	Latch name
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
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 3-69

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

Column	Datatype	Description
WHERE	VARCHAR2(64)	This column is obsolete and maintained only for compatibility reasons. 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	
LONGHOLD_COUNT	NUMBER	
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 3-69

V$LATCHHOLDER

This view contains information about the current latch holders.

Column	Datatype	Description
PID	NUMBER	Identifier of process holding the latch
SID	NUMBER	Identifier of the session that owns the latch
LADDR	RAW(4)	Latch address
NAME	VARCHAR2	Name of 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(64)	Latch name

See Also: "V$LATCH" on page 3-69

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 be comprised 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)	The library cache namespace
GETS	NUMBER	The number of times a lock was requested for objects of this namespace
GETHITS	NUMBER	The number of times an object's handle was found in memory
GETHITRATIO	NUMBER	The ratio of GETHITS to GETS
PINS	NUMBER	The number of times a PIN was requested for objects of this namespace
PINHITS	NUMBER	The number of times all of the metadata pieces of the library object were found in memory
PINHITRATIO	NUMBER	The 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	The number of GET requests lock instance locks

Column	Datatype	Description
DLM_PIN_REQUESTS	NUMBER	The number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER	The number of release requests PIN instance locks
DLM_INVALIDATION_REQUESTS	NUMBER	The number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER	The 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	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 server and outstanding requests for a lock or latch.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
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 3–1.
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: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S / Row-X (SSX) ■ 6 - exclusive (X)

Column	Datatype	Description
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 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 3–1 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 commit 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
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

Table 3–1 (Cont.) Values for the TYPE Column: System Types

System Type	Description	System Type	Description
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

This is an Oracle9*i* Real Application Clusters view. V$LOCK_ACTIVITY displays the DLM lock operation activity of the current instance. Each row corresponds to a type of lock operation.

See Also: *Oracle9i Real Application Clusters Concepts*

Column	Datatype	Description
FROM_VAL	VARCHAR2(4)	PCM lock initial state: NULL; S; X; SSX
TO_VAL	VARCHAR2(4)	PCM lock initial state: NULL; S; X; SSX
ACTION_VAL	VARCHAR2(51)	Description of lock conversions: ■ Lock buffers for read ■ Lock buffers for write ■ Make buffers CR (no write) ■ Upgrade read lock to write ■ Make buffers CR (write dirty buffers) ■ Downgrade write lock to read (write dirty buffers) ■ Write transaction table / undo blocks ■ Transaction table / undo blocks (write dirty buffers) ■ Make transaction table / undo blocks available share ■ Rearm transaction table write mechanism
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(15)	OS user name
PROCESS	VARCHAR2(9)	OS process ID
LOCKED_MODE	NUMBER	Lock mode

V$LOG

This view contains log file information from the control files.

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
ARCHIVED	VARCHAR2	Archive status (YES | NO)

Column	Datatype	Description
STATUS	VARCHAR2(16)	Log status: ■ UNUSED - The 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 - This is the current redo log. This implies that the redo log is active. The redo log could be open or closed. ■ ACTIVE - The log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It might or might not be archived. ■ CLEARING - The 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 - The 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 - The log is no longer needed for instance recovery. It may be in use for media recovery. It might or might not be archived.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log

V$LOG_HISTORY

This view contains log history information from the control file.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
FIRST_TIME	DATE	Time of first entry (lowest SCN) in the log. This column was previously named TIME.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log. This column was previously named LOW_CHANGE#.
NEXT_CHANGE#	NUMBER	Highest SCN in the log. This column was previously named HIGH_CHANGE#.
RECID	NUMBER	Control file record ID
STAMP	NUMBER	Control file record stamp

V$LOGFILE

This view contains information about redo log files.

Column	Datatype	Description
GROUP#	NUMBER	Redo log group identifier number
STATUS	VARCHAR2	Status of this log member: INVALID (file is inaccessible), STALE (file's contents are incomplete), DELETED (file is no longer used), or blank (file is in use)
MEMBER	VARCHAR2	Redo log member name

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 3-79

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	
TIMESTAMP	DATE	Timestamp
COMMIT_TIMESTAMP	DATE	
THREAD#	NUMBER	Number of the thread which generated the redo record

Column	Datatype	Description
LOG_ID	NUMBER	Number of the log file which contains the redo record. This number is generated by LogMiner to uniquely identify a log, and corresponds to the LOG_ID column in V$LOGMNR_FILES.
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
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_DOBJ#	NUMBER	Data block data object number identifying the object within the tablespace
SEG_OWNER	VARCHAR2(30)	Owner of the segment
SEG_NAME	VARCHAR2(81)	Name of the segment
SEG_TYPE	NUMBER	Segment type
SEG_TYPE_NAME	VARCHAR2(32)	Segment type name
TABLE_SPACE	VARCHAR2(30)	Tablespace name
ROW_ID	VARCHAR2(18)	Row ID
SESSION#	NUMBER	Session number of the session which generated the redo. A NULL value is reported if the session number is not available from the redo log.
SERIAL#	NUMBER	Serial number of the session which generated the redo. The session number and serial # can be used to uniquely identify the Oracle session. A NULL value is reported if the session number is not available from the redo log.
USERNAME	VARCHAR2(30)	The user name

Column	Datatype	Description
SESSION_INFO	VARCHAR2(4000)	Session information
ROLLBACK	NUMBER	The rollback request
OPERATION	VARCHAR2(30)	The operation
OPERATION_CODE	NUMBER	The operation code
SQL_REDO	VARCHAR2(4000)	SQL redo
SQL_UNDO	VARCHAR2(4000)	SQL undo
RS_ID	VARCHAR2(30)	Record set ID
SSN	NUMBER	SQL sequence number
CSF	NUMBER	Continuation SQL flag
INFO	VARCHAR2(32)	Informational message
STATUS	VARCHAR2 (16)	The status
REDO_VALUE	RAW(4)	
UNDO_VALUE	RAW(4)	
SQL_COLUMN_TYPE	VARCHAR2(32)	Type of data in the SQL_REDO and SQL_UNDO columns
SQL_COLUMN_NAME	VARCHAR2(32)	The name of the LOB or LONG columns
REDO_LENGTH	NUMBER	The length of the data value in the SQL_REDO column.
REDO_OFFSET	NUMBER	The length of the data value in the SQL_REDO column
UNDO_LENGTH	NUMBER	The length of the data value in the SQL_UNDO column
UNDO_OFFSET	NUMBER	If SQL_COLUMN_TYPE is LOB_COLUMN or LONG_COLUMN then this column contains the byte offset of the SQL_UNDO column data in the column indicated by SQL_COLUMN_NAME. Zero by default.

V$LOGMNR_DICTIONARY

This view contains log history information.

Column	Datatype	Description
TIMESTAMP	DATE	Date the dictionary was created
DB_ID	NUMBER	Database ID
DB_NAME	VARCHAR2(8)	Name of the database
FILENAME	VARCHAR2(513)	Dictionary filename
DICTIONARY_SCN	NUMBER	System change number when the dictionary was created
RESET_SCN	NUMBER	Reset log SCN when the dictionary was created

Column	Datatype	Description
RESET_SCN_TIME	DATE	Time when the reset log SCN was obtained to create the dictionary
ENABLED_THREAD_MAP	RAW(16)	Bit map of currently enabled threads when the dictionary was created
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	Identifies the log file.
FILENAME	VARCHAR2(513)	The filename
LOW_TIME	DATE	The oldest date of any records in the file
HIGH_TIME	DATE	The most recent date of any records in the file
DB_ID	NUMBER	The database ID
DB_NAME	VARCHAR2(8)	The name of the database
RESET_SCN	NUMBER	The reset log SCN when the log was created
RESET_SCN_TIME	NUMBER	The time when the reset log SCN was obtained to create the log
THREAD_ID	NUMBER	The thread number
THREAD_SQN	NUMBER	The 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.
INFO	VARCHAR2(32)	Informational message. A value of MISSING_LOGFILE will be assigned to a row entry where a needed logfile is missing from the list of log files.
STATUS	NUMBER	Indicates the status of a logfile. A NULL value indicates a valid logfile; a non-NULL value indicates further information is contained in the INFO column as a text string. All logfiles successfully added to the file list will have a status value of NULL.

V$LOGMNR_PARAMETERS

This view contains log information.

Column	Datatype	Description
START_DATE	DATE	The date to start search at
END_DATE	DATE	The date to end search at
START_SCN	NUMBER	The system change number to start search
END_SCN	NUMBER	The system change number to end search
INFO	VARCHAR2(32)	An informational message
STATUS	NUMBER	The status. A NULL value indicates parameters are valid. A non-NULL value indicates further information is contained in the INFO column as a text string.

V$LOGSTDBY

V$LOGSTDBY provides dynamic information about what is happening to the 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.

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
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 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 while log apply services are running. If log apply services are not running, then the values for the statistics are cleared.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the statistic, state, or status: **Note:** Many of the following statistics are subject to change or deletion; programmers should write application code to tolerate missing or extra statistics. ■ number of preparers ■ number of appliers ■ maximum SGA for LCR cache ■ parallel servers in use ■ transaction consistency ■ coodinator 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
VALUE	VARCHAR2(64)	Value of the statistic or state information

V$MANAGED_STANDBY

V$MANAGED_STANDBY displays current and status information for some Oracle database server processes related to the Data Guard environment for physical standby database only.

Column	Datatype	Description
PROCESS	VARCHAR2(7)	Type of process whose information is being reported. Possible values are: ■ RFS - remote file server ■ MRP0 - detached recovery server process ■ MR(fg) - foreground recovery session
PID	NUMBER	Operating system process identifier of process
STATUS	VARCHAR2(12)	Current process status. Possible values are: ■ UNUSED - no active process ■ ALLOCATED - process is active but not currently connected to a primary database client ■ CONNECTED - network connection established to a primary database client ■ ATTACHED - process is actively attached and communicating to a primary database client ■ 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 archived redo log data ■ 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. Possible values are: ■ ARCHIVAL - foreground (manual) archival process (SQL) ■ ARCH - background ARC*n* 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

Column	Datatype	Description
THREAD#	NUMBER	Archived redo log thread number
SEQUENCE#	NUMBER	Archived redo log sequence number
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: ■ 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$MAP_FILE)
EXT_NUM	NUMBER	File extent number
EXT_ELEM_OFF	NUMBER	Element offset in HKB
EXT_SIZE	NUMBER	File extent size in HKB
EXT_FILE_OFF	NUMBER	File Offset in HKB
EXT_TYPE	VARCHAR2(6)	File Extent Type (DATA, PARITY, or NONE)
ELEM_IDX	NUMBER	Index in V$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

Column	Datatype	Description
PATH_NAME	VARCHAR2(1024)	Path name
MAP_FILE	VARCHAR2(1)	Indicates whether this library supports mapping files (Y) or not (N)
FILE_CFGID	VARCHAR2(13)	Type of configuration ID supported for files: ■ 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: ■ 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 mappping 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$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
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 (ON, READY, or OFF)
DIRTY_LIMIT	NUMBER	Dirty buffer limit derived from the MTTR being simulated
ESTD_CACHE_WRITES	NUMBER	Estimated number of cache physical writes under this MTTR
ESTD_CACHE_WRITE_FACTOR	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.
ESTD_TOTAL_WRITES	NUMBER	Estimated total number of physical writes under this MTTR
ESTD_TOTAL_WRITE_FACTOR	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.
ESTD_TOTAL_IOS	NUMBER	Estimated total number of I/Os under this MTTR
ESTD_TOTAL_IO_FACTOR	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$MYSTAT

This view contains statistics on the current session.

Column	Datatype	Description
SID	NUMBER	The ID of the current session
STATISTIC#	NUMBER	The number of the statistic
VALUE	NUMBER	The value of the statistic

V$NLS_PARAMETERS

This view contains current values of NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2	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	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$SESSION" on page 3-123 and "V$SQL" on page 3-139

Column	Datatype	Description
FROM_ADDRESS	RAW(4)	The address of a procedure, package, or cursor that is currently loaded in the shared pool
FROM_HASH	NUMBER	The hash value of a procedure, package, or cursor that is currently loaded in the shared pool
TO_OWNER	VARCHAR2(64)	The owner of the object that is depended on
TO_NAME	VARCHAR2(1000)	The name of the object that is depended on
TO_ADDRESS	RAW(4)	The 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.

Column	Datatype	Description
TO_HASH	NUMBER	The 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	The 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)	The index name in sys.obj$.name
TABLE_NAME	VARCHAR2(30)	The table name in sys.obj$.name
MONITORING	VARCHAR2(3)	YES | NO
USED	VARCHAR2(3)	YES | NO
START_ MONITORING	VARCHAR2(19)	The start monitoring time in sys.object_stats.start_monitoring
END_MONITORING	VARCHAR2(19)	The end monitoring time in sys.object_stats.end_monitoring

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 3-37

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

V$OPEN_CURSOR

This view lists cursors that each user session currently has opened and parsed.

Column	Datatype	Description
SADDR	RAW	Session address
SID	NUMBER	Session identifier
USER_NAME	VARCHAR2(30)	User that is logged in to the session
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely the SQL statement being executed in the session
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely 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 server.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	The name of the option
VALUE	VARCHAR2(64)	TRUE if the option is installed

V$PARALLEL_DEGREE_LIMIT_MTH

This view displays all available parallel degree limit resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	The 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(64)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 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
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: ■ 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.

Column	Datatype	Description
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: ■ 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)
DESCRIPTION	VARCHAR2(64)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update

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(64)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 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

Column	Datatype	Description
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: ■ 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.
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: ■ 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)
DESCRIPTION	VARCHAR2(64)	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$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: *Oracle9i Database Performance Tuning Guide and Reference* 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: *Oracle9i Database Performance Tuning Guide and Reference* 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
VALUE	NUMBER	Statistic value
UNITS	VARCHAR2(12)	Unit for the value (microseconds, bytes, or percent)

V$PQ_SESSTAT

This view lists session statistics for parallel queries.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	The following statistics (fixed rows) have been defined for this view. 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. Name of the statistic: ■ Queries Parallelized - number of queries 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	The value of the statistic for the last operation
SESSION_TOTAL	NUMBER	The 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)	The current status of the parallel execution server (BUSY | IDLE)
SESSIONS	NUMBER	The number of sessions that have used this parallel execution server
IDLE_TIME_CUR	NUMBER	The amount of time spent idle while processing statements in the current session
BUSY_TIME_CUR	NUMBER	The amount of time spent busy while processing statements in the current session
CPU_SECS_CUR	NUMBER	The amount of CPU time spent on the current session
MSGS_SENT_CUR	NUMBER	The number of messages sent while processing statements for the current session
MSGS_RCVD_CUR	NUMBER	The number of messages received while processing statements for the current session
IDLE_TIME_TOTAL	NUMBER	The total amount of time this query server has been idle
BUSY_TIME_TOTAL	NUMBER	The total amount of time this query server has been active
CPU_SECS_TOTAL	NUMBER	The total amount of CPU time this query server has used to process statements
MSGS_SENT_TOTAL	NUMBER	The total number of messages this query server has sent
MSGS_RCVD_TOTAL	NUMBER	The total number of messages this query server has received

V$PQ_SYSSTAT

This view lists system statistics for parallel queries.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	The following statistics (fixed rows) have been defined for this view. 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
		Name of the statistic: ■ 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 ■ 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	The 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	The data flow operator (DFO) tree number to differentiate queries
TQ_ID	NUMBER	The 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

Column	Datatype	Description
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)	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$PROXY_ARCHIVEDLOG

This view contains descriptions of archived log backups which are taken with a new feature called 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; 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	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.
STATUS	VARCHAR2(1)	Indicates the status of the copy: ■ A - The object is available ■ D - The object is deleted ■ X - The 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.
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
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	The starting time
COMPLETION_TIME	DATE	The completion time

Column	Datatype	Description
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds

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; not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the copy 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
TAG	VARCHAR2(32)	Proxy copy tag
STATUS	VARCHAR2(1)	Indicates the status of the copy: ■ A - The object is available ■ D - The object is deleted ■ X - The 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.
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	The highest change in any block of the file, if known

Column	Datatype	Description
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	0 if this backup is part of an incremental backup strategy, otherwise NULL
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 (ie, 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	The starting time
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The 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)	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$PWFILE_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

Column	Datatype	Description
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(9)	The OS process ID
SID	NUMBER	The session ID of slave, if in use
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)	The name of the statistic: ■ Servers In Use - the number of PX servers currently performing parallel operations ■ Servers Available - the number of PX servers available to perform parallel operations ■ Servers Started - the number of times the system has had to create a PX server process ■ Server Shutdown - The number of times a PX server process has been shutdown. A PX server process will be shutdown if it has not been used recently. The length of time it may remain "Available" is controlled by the initialization parameter PARALLEL_SERVER_IDLE_TIME. If this value is large, consider increasing the parameter. This will improve performance by avoiding the latency of PX server process creation.

Column	Datatype	Description
		■ Servers HWM - the maximum number of concurrent PX server processes If this number is equal to the initialization parameter PARALLEL_MAX_SERVERS, consider increasing the parameter. This could allow you to increase your throughput, especially if your system is under-utilized and the V$SYSSTAT statistic "Parallel operations downgraded to serial" is large. ■ Servers Cleaned Up - The 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, you should determine the cause. ■ Sessions - the total number of sessions created by all PX servers ■ Memory Chunks Allocs - the number of large memory chunks allocated by PX servers ■ Memory Chunks Freed - the number of large memory chunks freed ■ Memory Chunks Current - the number of large memory chunks currently being used ■ Memory Chunks HWM - the maximum number of concurrently allocated chunks ■ Buffers allocated - the number of times a message buffer has been allocated ■ Buffers freed - the number of times a message buffer has been freed ■ Buffers Current - the number of message buffers currently being used ■ Buffers HWM - the maximum number of concurrently allocated message buffers
VALUE	NUMBER	The value of the statistic

V$PX_SESSION

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4)	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

Column	Datatype	Description
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)	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 (identifier)
VALUE	NUMBER	Statistic value

V$QUEUE

This view contains information on the multi-thread message queues.

Column	Datatype	Description
PADDR	RAW(4)	Address of the process that owns the queue
TYPE	VARCHAR2	Type of queue (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. Divide by TOTALQ for average wait per item.
TOTALQ	NUMBER	Total number of items that have ever been in the queue

V$QUEUEING_MTH

This view 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 only for compatibility reasons. 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_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.

See Also: *Oracle9i Recovery Manager User's Guide*

Column	Datatype	Description
FILENUM	NUMBER	The number of the file being recovered
FILENAME	VARCHAR2(257)	The filename of the datafile being recovered
STATUS	VARCHAR2(13)	The status of the recovery (IN RECOVERY|CURRENT| NOT RECOVERED)

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.

See Also: "V$LOG_HISTORY" on page 3-79 and *Oracle9i Recovery Manager User's Guide*

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
TIME	VARCHAR2	Time of first entry (lowest SCN) in the log
ARCHIVE_NAME	VARCHAR2	Name of the file when archived, using the naming convention specified by "LOG_ARCHIVE_FORMAT"

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.

See Also: *Oracle9i User-Managed Backup and Recovery Guide*

Column	Datatype	Description
TYPE	VARCHAR2(64)	The type of recovery operation being performed
ITEM	VARCHAR2(32)	The item being measured
SOFAR	NUMBER	The amount of work done so far
TOTAL	NUMBER	The total amount of work expected

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.

See Also: *Oracle9i Recovery Manager User's Guide*

Column	Datatype	Description
RECOVERY_CHECKPOINT	DATE	The 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	The 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)	The 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(257)	The filename of the log
PREVIOUS_LOG_STATUS	VARCHAR2(13)	The status of the previous log. Contains one of the following values: RELEASE; WRONG NAME; MISSING NAME; UNNEEDED NAME; NONE
REASON	VARCHAR2(13)	The reason recovery is returning control to the user (NEED LOG|LOG REUSED|THREAD DISABLED)

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(64)	The name of the keyword
LENGTH	NUMBER	The length of the keyword

V$RESOURCE

This view contains resource name and address information.

Column	Datatype	Description
ADDR	RAW(4)	Address of resource object
TYPE	VARCHAR2	Resource type; the resource types are listed in Table 3–1 on page 3-76
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 3–2.

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 3–2)
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
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 3–2 Values for RESOURCE_NAME column

Resource Name	Corresponds to this Initialization Parameter
DML_LOCKS	See "DML_LOCKS" on page 1-53
ENQUEUE_LOCKS	This value is computed by Oracle. Use the V$ENQUEUE_LOCK view (described on on page 3-51) to obtain more information about the enqueue locks.
ENQUEUE_RESOURCES	See "ENQUEUE_RESOURCES" on page 1-55
LM_PROCESSES	Lock manager processes
LM_LOCKS	See "LOCAL_LISTENER" on page 1-73
MAX_SHARED_SERVERS	See "MAX_SHARED_SERVERS" on page 1-94
PARALLEL_MAX_SERVERS	See "PARALLEL_MAX_SERVERS" on page 1-123
PROCESSES	See "PROCESSES" on page 1-133
ROLLBACK_SEGMENTS	See "MAX_ROLLBACK_SEGMENTS" on page 1-94
SESSIONS	See "SESSIONS" on page 1-148

Table 3–2 (Cont.) Values for RESOURCE_NAME column

Resource Name	Corresponds to this Initialization Parameter
SORT_SEGMENT_LOCKS	This value is computed by Oracle
TEMPORARY_LOCKS	This value is computed by Oracle
TRANSACTIONS	See "TRANSACTIONS" on page 1-164

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)	The name for this particular configuration. Example: RETENTION POLICY
VALUE	VARCHAR2(1025)	The value for this configuration row. Example: RETENTION POLICY TO RECOVERY WINDOW OF 10 DAYS

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	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:** *Oracle9i Database Administrator's Guide.*

Column	Datatype	Description
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 water mark 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
AVEACTIVE	NUMBER	Current size of active extents, averaged over time.
STATUS	VARCHAR2(15)	Rollback segment status: ■ 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	Parent or subordinate row cache type
SUBORDINATE#	NUMBER	Subordinate set number
PARAMETER	VARCHAR2	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

Column	Datatype	Description
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	The number of DLM requests
DLM_CONFLICTS	NUMBER	The number of DLM conflicts
DLM_RELEASES	NUMBER	The 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
INDX	NUMBER	Index of the row
HASH	NUMBER	The hash value
ADDRESS	RAW(4)	The address of the parent object
CACHE#	NUMBER	The parent cache ID
CACHE_NAME	VARCHAR2(64)	The parent cache name
EXISTENT	VARCHAR2(1)	Whether the object is an existing object
LOCK_MODE	NUMBER	The mode the lock is held in
LOCK_REQUEST	NUMBER	The mode the lock is requested in
TXN	RAW(4)	The transaction currently locking the object
SADDR	RAW(4)	The address of the session
Note: The following are relevant only for Oracle Real Application Clusters.		
INST_LOCK_REQUEST	NUMBER	The 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)	The type of instance lock

Column	Datatype	Description
INST_LOCK_ID1	RAW(4)	The ID associated with the instance lock
INST_LOCK_ID2	RAW(4)	The ID associated with the instance lock
KEY	RAW(100)	The contents of the key

V$ROWCACHE_SUBORDINATE

This view displays information for subordinate objects in the data dictionary.

Column	Datatype	Description
INDX	NUMBER	The index
HASH	NUMBER	The hash value
ADDRESS	RAW(4)	The 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)	The 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.

See Also:

- *Oracle9i Database Administrator's Guide* for information on resource groups
- *Oracle9i Supplied PL/SQL Packages and Types Reference* for information on creating resource groups with the DBMS_RESOURCE_MANAGER package

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the consumer group
ACTIVE_SESSIONS	NUMBER	Number of currently active sessions in this consumer group

Column	Datatype	Description
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 this 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 this consumer group had to wait for CPU
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in this consumer group
YIELDS	NUMBER	Cumulative number of times sessions in this 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

V$RSRC_CONSUMER_GROUP_CPU_MTH

This view lists all resource allocation methods defined for resource consumer groups.

See Also:

- *Oracle9i Database Administrator's Guide* for information on resource allocation methods
- *Oracle9i Supplied 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 3-121 for a listing of all resource allocation methods defined for resource plans

Column	Datatype	Description
NAME	VARCHAR2(40)	The 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 2-201 for a listing of all plans in the database
- *Oracle9i Database Administrator's Guide* for information on resource plans
- *Oracle9i Supplied 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

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 3-120 for a listing of resource allocation methods defined for consumer groups
- *Oracle9i Database Administrator's Guide* for information on resource plans
- *Oracle9i Supplied 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$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
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$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$SESSION

This view lists session information for each current session.

Column	Datatype	Description
SADDR	RAW(4)	Session address
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 and another session begins with the same session ID.
AUDSID	NUMBER	Auditing session ID
PADDR	RAW(4)	Address of the process that owns this 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 3–3. These values also appear in the data dictionary view AUDIT_ACTIONS.
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: ACTIVE (currently executing SQL), INACTIVE, KILLED (marked to be killed), CACHED (temporarily cached for use by Oracle*XA), SNIPED (session inactive, waiting on the client)

Column	Datatype	Description
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(9)	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)	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
PREV_SQL_ADDR	RAW(4)	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
MODULE	VARCHAR2(48)	Contains the name of the currently executing module as set by calling the DBMS_APPLICATION_INFO.SET_MODULE procedure
MODULE_HASH	NUMBER	The hash value of the above MODULE
ACTION	VARCHAR2(32)	Contains the name of the currently executing action as set by calling the DBMS_APPLICATION_INFO.SET_ACTION procedure
ACTION_HASH	NUMBER	The 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.
ROW_WAIT_OBJ#	NUMBER	Object ID for the table containing the ROWID specified in ROW_WAIT_ROW#
ROW_WAIT_FILE#	NUMBER	Identifier for the datafile containing the ROWID 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 ROWID 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.

Column	Datatype	Description
ROW_WAIT_ROW#	NUMBER	The current ROWID 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	The 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: ■ NONE - failover is disabled for this session ■ SESSION - the client is able to fail over its session following a disconnect ■ SELECT - the client is able to fail over queries in progress as well **See Also:** ■ *Oracle9i Database Concepts* for more information on TAF ■ *Oracle9i Net Services Administrator's Guide* for information on configuring TAF
FAILOVER_METHOD	VARCHAR2(10)	Indicates the transparent application failover method for the session: ■ NONE - failover is disabled for this session ■ BASIC - the client itself reconnects following a disconnect ■ PRECONNECT - the backup instance can support all connections from every instance for which it is backup
FAILED_OVER	VARCHAR2(3)	Indicates (YES | NO) whether the session is running in failover mode and failover has occurred
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, value is 0.
CLIENT_IDENTIFIER	VARCHAR2(64)	

Table 3–3 Numbers Displayed in COMMAND Column of V$SESSION and Corresponding Commands

1: CREATE TABLE	31: NOAUDIT	61: DROP TRIGGER
2: INSERT	32: CREATE DATABASE LINK	62: ANALYZE TABLE
3: SELECT	33: DROP DATABASE LINK	63: ANALYZE INDEX
4: CREATE CLUSTER	34: CREATE DATABASE	64: ANALYZE CLUSTER
5: ALTER CLUSTER	35: ALTER DATABASE	65: CREATE PROFILE
6: UPDATE	36: CREATE ROLLBACK SEGMENT	66: DROP PROFILE
7: DELETE	37: ALTER ROLLBACK SEGMENT	67: ALTER PROFILE
8: DROP CLUSTER	38: DROP ROLLBACK SEGMENT	68: DROP PROCEDURE
9: CREATE INDEX	39: CREATE TABLESPACE	69: DROP PROCEDURE
10: DROP INDEX	40: ALTER TABLESPACE	70: ALTER RESOURCE COST
11: ALTER INDEX	41: DROP TABLESPACE	71: CREATE SNAPSHOT LOG
12: DROP TABLE	42: ALTER SESSION	72: ALTER SNAPSHOT LOG
13: CREATE SEQUENCE	43: ALTER USE	73: DROP SNAPSHOT LOG
14: ALTER SEQUENCE	44: COMMIT	74: CREATE SNAPSHOT
15: ALTER TABLE	45: ROLLBACK	75: ALTER SNAPSHOT
16: DROP SEQUENCE	46: SAVEPOINT	76: DROP SNAPSHOT
17: GRANT	47: PL/SQL EXECUTE	79: ALTER ROLE
18: REVOKE	48: SET TRANSACTION	85: TRUNCATE TABLE
19: CREATE SYNONYM	49: ALTER SYSTEM SWITCH LOG	86: TRUNCATE COUSTER
20: DROP SYNONYM	50: EXPLAIN	88: ALTER VIEW
21: CREATE VIEW	51: CREATE USER	91: CREATE FUNCTION
22: DROP VIEW	25: CREATE ROLE	92: ALTER FUNCTION
23: VALIDATE INDEX	53: DROP USER	93: DROP FUNCTION
24: CREATE PROCEDURE	54: DROP ROLE	94: CREATE PACKAGE
25: ALTER PROCEDURE	55: SET ROLE	95: ALTER PACKAGE
26: LOCK TABLE	56: CREATE SCHEMA	96: DROP PACKAGE
27: NO OPERATION	57: CREATE CONTROL FILE	97: CREATE PACKAGE BODY
28: RENAME	58: ALTER TRACING	98: ALTER PACKAGE BODY
29: COMMENT	59: CREATE TRIGGER	99: DROP PACKAGE BODY
30: AUDIT	60: ALTER TRIGGER	

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(15)	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
OSUSER	VARCHAR2(30)	The external username for this database user
NETWORK_SERVICE_BANNER	VARCHAR2(2000)	Product banners for each Net8 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 1-162

Column	Datatype	Description
SID	NUMBER	The ID of the session
EVENT	VARCHAR2(64)	The name of the wait event **See Also:** Appendix A, "Oracle Wait Events"
TOTAL_WAITS	NUMBER	The total number of waits for this event by this session
TOTAL_TIMEOUTS	NUMBER	The total number of timeouts for this event by this session
TIME_WAITED	NUMBER	The total amount of time waited for this event by this session, in hundredths of a second
AVERAGE_WAIT	NUMBER	The average amount of time waited for this event by this session, in hundredths of a second
MAX_WAIT	NUMBER	The maximum time (in hundredths of a second) waited for this event by this session

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: *Oracle9i Supplied 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)	Used with the value of the SQL_HASH_VALUE column to identify the SQL statement associated with the operation
SQL_HASH_VALUE	NUMER	Used with the value of the SQL_ADDRESS column to identify 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

Column	Datatype	Description
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 HITS/PINS
TRUE_HIT_RATIO	NUMBER	The ratio of TRUE_HITS/PINS
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

This view lists 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, you must set the TIMED_STATISTICS 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.

See Also: "TIMED_STATISTICS" on page 1-162 and Appendix A, "Oracle Wait Events"

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 A, "Oracle Wait Events"
P1TEXT	VARCHAR2	Description of first additional parameter
P1	NUMBER	First additional parameter
P1RAW	RAW(4)	First additional parameter
P2TEXT	VARCHAR2	Description of second parameter
P2	NUMBER	Second additional parameter
P2RAW	RAW(4)	Second additional parameter
P3TEXT	VARCHAR2	Description of third parameter
P3	NUMBER	Third additional parameter
P3RAW	RAW(4)	Third additional parameter
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	The seconds in wait
STATE	VARCHAR2	Wait state: ■ 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)

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.

See Also: "V$STATNAME" on page 3-159 and Appendix C, "Statistics Descriptions"

Column	Datatype	Description
SID	NUMBER	Session identifier
STATISTIC#	NUMBER	Statistic number (identifier) **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

V$SGA

This view contains summary information on the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2	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(6)	Operation type: ■ GROW ■ SHRINK
OPER_MODE	VARCHAR2(6)	Operation mode: ■ MANUAL ■ AUTO
PARAMETER	VARCHAR2(64)	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_RESIZE_OPS

V$SGA_RESIZE_OPS displays information about the last 100 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(6)	Operation type: ■ GROW ■ SHRINK
OPER_MODE	VARCHAR2(6)	Operation mode: ■ MANUAL ■ AUTO
PARAMETER	VARCHAR2(64)	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(6)	Completion status of the operation: ■ NORMAL ■ CANCEL ■ ERROR
START_TIME	DATE	Start time of the operation
END_TIME	DATE	End time of 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

Column	Datatype	Description
OPER_COUNT	NUMBER	Number of operations since instance startup
LAST_OPER_TYPE	VARCHAR2(6)	Last completed operation for the component: ■ GROW ■ SHRINK
LAST_OPER_MODE	VARCHAR2(6)	Mode of the last completed operation: ■ MANUAL ■ AUTO
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$SGASTAT

This view contains detailed information on the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2(26)	SGA component name
BYTES	NUMBER	Memory size in bytes
POOL	VARCHAR2(11)	Designates the pool in which the memory in NAME resides: ■ large pool - memory is allocated from the large pool ■ shared pool - memory is allocated from the shared pool ■ java pool - memory is allocated from the Java pool

V$SHARED_POOL_ADVICE

V$SHARED_POOL_ADVICE displays information about estimated parse time savings in the shared pool for different sizes. The sizes range from 50% 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_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 1-150		
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

Column	Datatype	Description
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	Name of the server
PADDR	RAW(4)	Server's process address
STATUS	VARCHAR2	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)	Address of circuit currently being serviced
IDLE	NUMBER	Total idle time in hundredths of a second

Column	Datatype	Description
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 CIRCUITS initialization parameter, then consider raising the value of CIRCUITS. **See Also:** "CIRCUITS" on page 1-24
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 SHARED_SERVER_SESSIONS initialization parameter, then consider raising the value of SHARED_SERVER_SESSIONS. **See Also:** "SHARED_SERVER_SESSIONS" on page 1-151
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 MAX_SHARED_SERVERS initialization parameter, then consider raising the value of SHARED_SERVERS. **See Also:** "SHARED_SERVERS" on page 1-152

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
CURRENT_USERS	NUMBER	Number of active users of the segment

Column	Datatype	Description
TOTAL_EXTENTS	NUMBER	Total number of extents in the segment
TOTAL_BLOCKS	NUMBER	Total number of blocks in the segment
RELATIVE_FNO	NUMBER	Relative file number of the sort segment header
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

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)
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.

Column	Datatype	Description
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
SHARABLE_MEM	NUMBER	Amount of shared memory used by this child cursor (in bytes)
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of this child cursor (in bytes)
RUNTIME_MEM	NUMBER	Fixed amount of memory required during the execution of this child cursor
SORTS	NUMBER	Number of sorts that were done for this 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
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
BUFFER_GETS	NUMBER	Number of buffer gets for this child cursor
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
PARSING_USER_ID	NUMBER	User ID of the user who originally built this child cursor

Column	Datatype	Description
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
ADDRESS	RAW(4)	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
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)	Address of the child cursor
SQLTYPE	NUMBER	Denotes the version of the SQL language used for this statement
REMOTE	VARCHAR2(1)	(Y / N) Identifies whether the cursor is remote mapped or not
OBJECT_STATUS	VARCHAR2(19)	Status of the cursor (VALID / 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)	

Column	Datatype	Description
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

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)	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)	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)	Pointer to the parent cursor handle
PARENT_LOCK	RAW(4)	Pointer to the parent cursor lock
CHILD_LOCK	RAW(4)	Pointer to the child cursor lock
CHILD_PIN	RAW(4)	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

Column	Datatype	Description
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_PLAN

V$SQL_PLAN contains the execution plan information for each child cursor loaded in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4)	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$SQLAREA to add the cursor-specific information.
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$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(10)	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(64)	Name of the table or index
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.

Column	Datatype	Description
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.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them

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)	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$SQLAREA to locate the parent cursor.
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$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
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)	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$SQLAREA to add the cursor-specific information.
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$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(10)	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(64)	Name of the table or index
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)

Column	Datatype	Description
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.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them
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.

Column	Datatype	Description
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
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: ■ 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

Column	Datatype	Description
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)	KGL address
PARENT_HANDLE	RAW(4)	KGL address of parent
HASH_VALUE	NUMBER	SQL hash
CHILD_NUMBER	NUMBER	Number of child (instance) for hash
PARSING_USER_ID	NUMBER	
PARSING_SCHEMA_ID	NUMBER	
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
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
ADDRESS	RAW(4)	Address of the child cursor
KGLHDPAR	RAW(4)	Address of the parent cursor
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

Column	Datatype	Description
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)	The SQL text of the shared cursor child object for which this row is displaying information
HASH_VALUE	NUMBER	The hash value of the above SQL text in the shared pool
HEAP_DESC	RAW(4)	The 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
COMMENT	VARCHAR2(16)	The whole comment field that was supplied when this memory chunk was allocated
CHUNK_PTR	RAW(4)	The starting address of the allocated memory chunk
CHUNK_SIZE	NUMBER	The 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 recreate this chunk of memory should it need to be removed from the shared pool based on an LRU algorithm
SUBHEAP_DESC	RAW(4)	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)	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$SQLAREA to locate the parent cursor.
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns PARENT_HANDLE, HASH_VALUE, and CHILD_NUMBER can be used to join with V$SQL to locate the child cursor using this area.
WORKAREA_ADDRESS	RAW(4)	Address of the workarea 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

Column	Datatype	Description
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
WORKAREA_ADDRESS	RAW(4)	Address of the workarea 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.

Column	Datatype	Description
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: *Oracle9i Database Performance Tuning Guide and Reference* 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	Nnumber 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: *Oracle9i Database Performance Tuning Guide and Reference* 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
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.

Column	Datatype	Description
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	The 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
USERS_EXECUTING	NUMBER	Total number of users executing the statement over all child cursors
LOADS	NUMBER	The 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	The sum of all parse calls to all the child cursors under this parent
DISK_READS	NUMBER	The sum of the number of disk reads over all child cursors
BUFFER_GETS	NUMBER	The sum of buffer gets over all child cursors
ROWS_PROCESSED	NUMBER	The total number of rows processed on behalf of this SQL statement
COMMAND_TYPE	NUMBER	The Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement is executed
PARSING_USER_ID	NUMBER	The user ID of the user that has parsed the very first cursor under this parent
PARSING_SCHEMA_ID	NUMBER	The schema ID that was used to parse this child cursor
KEPT_VERSIONS	NUMBER	The number of child cursors that have been marked to be kept using the DBMS_SHARED_POOL package
ADDRESS	RAW(4)	The address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	The hash value of the parent statement in the library cache
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	The 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	The hash value of the action that is named in the ACTION column

Column	Datatype	Description
SERIALIZABLE_ ABORTS	NUMBER	Number of times the transaction fails to serialize, producing ORA-08177 errors, totalled over all the child cursors
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

V$SQLTEXT

This view contains the text of SQL statements belonging to shared SQL cursors in the SGA.

Column	Datatype	Description
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely a cached cursor
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2	A column containing one piece of the SQL text
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)

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.

See Also: "V$SQLTEXT" on page 3-157

Column	Datatype	Description
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely a cached cursor
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2	A column containing one piece of the SQL text
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)

V$STANDBY_LOG

V$STANDBY_LOG contains standby log file information from the control files.

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)
USED	NUMBER	Number of bytes used in the log
ARCHIVED	VARCHAR2(3)	Archive status
STATUS	VARCHAR2(10)	Indicates the log status. Possible values are: ■ 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 failure recovery. It may be in use for block recovery. It might or might 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 managed recovery. It might or might not be archived. ■ INVALIDATED - Archived the current redo log without a log switch.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log
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 lists the status of the statistics/advisories controlled by STATISTICS_LEVEL.

Column	Datatype	Description
STATISTICS_NAME	VARCHAR2(64)	Name of the statistics / advisory
DESCRIPTION	VARCHAR2(4000)	Description of the statistics / advisory
SESSION_STATUS	VARCHAR2(8)	(ENABLED / DISABLED) Status of the statistics / advisory for this session
SYSTEM_STATUS	VARCHAR2(8)	(ENABLED / DISABLED) Status of the statistics / advisory system-wide
ACTIVATION_LEVEL	VARCHAR2(7)	Indicates the level of STATISTICS_LEVEL that enables this statistics / advisory (BASIC, TYPICAL, or ALL)
STATISTICS_VIEW_NAME	VARCHAR2(64)	If there is a single view externalizing this statistics / advisory, then this column contains the name of that view. If there is no such view, then this column is empty. If there are multiple views involved, then the DESCRIPTION column mentions the view names.
SESSION_SETTABLE	VARCHAR2(3)	Indicates whether this statistics / 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.

See Also:

- "V$SESSTAT" on page 3-131 and "V$SYSSTAT" on page 3-164
- Appendix C, "Statistics Descriptions" for a description of each statistic
- Your operating system-specific Oracle documentation

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number

Column	Datatype	Description
		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	Statistic name
CLASS	NUMBER	A number representing one or more statistics class. The following class numbers are additive: ■ 1 - User ■ 2 - Redo ■ 4 - Enqueue ■ 8 - Cache ■ 16 - OS ■ 32 - Oracle Real Application Cluster ■ 64 - SQL ■ 128 - Debug

V$STREAMS_APPLY_COORDINATOR

V$STREAMS_APPLY_COORDINATOR monitors the overall performance of the coordinator. The coordinator creates and manages the 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 (INITIALIZING, APPLYING, SHUTTING DOWN CLEANLY, or 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
TOTAL_WAIT_DEPS	NUMBER	Number of dependency waits
TOTAL_WAIT_COMMITS	NUMBER	Number of waits to commit
TOTAL_ADMIN	NUMBER	Number of administrative requests issued
TOTAL_ASSIGNED	NUMBER	Number of transactions assigned
TOTAL_RECEIVED	NUMBER	Number of transactions received
TOTAL_ERRORS	NUMBER	Number of transactions which received an error on apply
LWM_TIME	DATE	Time low-watermark was updated

Column	Datatype	Description
LWM_MESSAGE_NUMBER	NUMBER	Number of the message corresponding to the low-watermark
LWM_MESSAGE_CREATE_TIME	DATE	Creation time of the message corresponding to the low-watermark
HWM_TIME	DATE	Time high-watermark was updated
HWM_MESSAGE_NUMBER	NUMBER	Highest message number
HWM_MESSAGE_CREATE_TIME	DATE	Creation time of the highest numbered message
STARTUP_TIME	DATE	Time when the apply started
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed scheduling messages

V$STREAMS_APPLY_READER

V$STREAMS_APPLY_READER displays information about the apply reader. The apply reader is a process which reads (dequeues) messages from the queue, computes message dependencies, builds transactions, and passes the transactions on to the coordinator in commit order for assignment to the apply servers.

Column	Datatype	Description
SID	NUMBER	Session ID of the apply server
SERIAL#	NUMBER	Serial number of the current 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
STATE	VARCHAR2(17)	State of the reader (IDLE, DEQUEUE MESSAGES, SCHEDULE MESSAGES)
TOTAL_MESSAGES_DEQUEUED	NUMBER	Total number of messages read
DEQUEUE_TIME	DATE	Time last message was received
DEQUEUED_MESSAGE_NUMBER	NUMBER	Number of the last message received
DEQUEUED_MESSAGE_CREATE_TIME	DATE	Creation time of the last message received
SGA_USED	NUMBER	Amount of SGA memory used
ELAPSED_DEQUEUE_TIME	NUMBER	time elapsed dequeuing messages
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed scheduling messages

V$STREAMS_APPLY_SERVER

V$STREAMS_APPLY_SERVER displays information about each apply server and its activities.

Column	Datatype	Description
SID	NUMBER	Session ID of the apply server
SERIAL#	NUMBER	Serial number of the current 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	Server number
STATE	VARCHAR2(20)	State of the server (IDLE, POLL SHUTDOWN, RECORD LOW-WATERMARK, ADD PARTITION, DROP PARTITION, EXECUTE TRANSACTION, WAIT COMMIT, WAIT DEPENDENCY, GET TRANSACTIONS, or WAIT FOR NEXT CHUNK)
XIDUSN	NUMBER	Transaction ID undo segment number currently being executed
XIDSLT	NUMBER	Transaction ID slot number currently being executed
XIDSQN	NUMBER	Transaction ID sequence number currently being executed
COMMITSCN	NUMBER	Commit number of the currently executing transaction
DEP_XIDUSN	NUMBER	Transaction ID undo segment number on which this server depends
DEP_XIDSLT	NUMBER	Transaction ID slot number on which this server depends
DEP_XIDSQN	NUMBER	Transaction ID sequence number on which this server depends
DEP_COMMITSCN	NUMBER	Commit number of the transaction which this server depends on
MESSAGE_SEQUENCE	NUMBER	Current message being applied by the server
TOTAL_ASSIGNED	NUMBER	Total number of transactions assigned to the server
TOTAL_ADMIN	NUMBER	Total number of administrative jobs done by the server
TOTAL_MESSAGES_APPLIED	NUMBER	Total number of messages applied by this server
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 of the last applied message
ELAPSED_DEQUEUE_TIME	NUMBER	Time elapsed dequeuing messages
ELAPSED_APPLY_TIME	NUMBER	Time elapsed applying messages

V$STREAMS_CAPTURE

V$STREAMS_CAPTURE displays information about the capture processes.

Column	Datatype	Description
SID	NUMBER	Session identifier of the capture process
SERIAL#	NUMBER	Session serial number of the capture process
CAPTURE#	NUMBER	Capture process number. A capture process is an Oracle background process, prefixed by CP.
CAPTURE_NAME	VARCHAR2(30)	Name of the capture process
STARTUP_TIME	DATE	Time when the capture started
STATE	VARCHAR2(17)	State of the capture process (INITIALIZING, CAPTURING CHANGES, EVALUATING RULE, ENQUEUING MESSAGE, SHUTTING DOWN, or CREATING LCR)
TOTAL_MESSAGES_CAPTURED	NUMBER	Total changes captured
CAPTURE_TIME	DATE	Time when the most recent message was captured
CAPTURE_MESSAGE_NUMBER	NUMBER	Number of the most recent message
CAPTURE_MESSAGE_CREATE_TIME	DATE	Creation time of the most recent message
TOTAL_MESSAGES_ENQUEUED	NUMBER	Total number of messages enqueued
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
ELAPSED_CAPTURE_TIME	NUMBER	Elapsed time capturing changes
ELAPSED_RULE_TIME	NUMBER	Elapsed time evaluating rules
ELAPSED_ENQUEUE_TIME	NUMBER	Elapsed time enqueuing messages
ELAPSED_LCR_TIME	NUMBER	Elapsed time creating lcrs

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$SYSSTAT

This view lists system statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V$STATNAME view.

See Also: "V$STATNAME" on page 3-159 and Appendix C, "Statistics Descriptions"

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: ■ 1 - User ■ 2 - Redo ■ 4 - Enqueue ■ 8 - Cache ■ 16 - OS ■ 32 - Oracle Real Application Cluster ■ 64 - SQL ■ 128 - Debug
VALUE	NUMBER	Statistic value

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 3-127

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 1-162

Column	Datatype	Description
EVENT	VARCHAR2(64)	The name of the wait event
TOTAL_WAITS	NUMBER	The total number of waits for this event
TOTAL_TIMEOUTS	NUMBER	The total number of timeouts for this event
TIME_WAITED	NUMBER	The total amount of time waited for this event, in hundredths of a second
AVERAGE_WAIT	NUMBER	The average amount of time waited for this event, in hundredths of a second

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(64)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	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)
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: ■ 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.
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)
DESCRIPTION	VARCHAR2(64)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update

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(64)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(512)	Parameter value
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: ■ 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.
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)

Column	Datatype	Description
DESCRIPTION	VARCHAR2(64)	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$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 this 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.

V$TEMP_CACHE_TRANSFER

V$TEMP_CACHE_TRANSFER displays the number of blocks pinged per datafile. This information in turn can be used to determine access patterns to existing datafiles and deciding new mappings from datafile blocks to PCM locks.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks in the file
X_2_NULL_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions
X_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the file was made STALE due to Shared-to-NULL conversions

Column	Datatype	Description
RBR	NUMBER	Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse block range cross instance calls for this file
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file

V$TEMP_EXTENT_MAP

This view displays the status of each unit for all temporary tablespaces.

Column	Datatype	Description
TABLESPACE_NAME	NUMBER	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
EXTENTS_USED	NUMBER	How many extents are actually being used

Column	Datatype	Description
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_SPACE_HEADER

This view displays aggregate information per file per 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)	The name of the temporary tablespace
FILE_ID	NUMBER	The 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	The absolute file number
CREATION_CHANGE#	NUMBER	The creation System Change Number (SCN)
CREATION_TIME	DATE	The creation time
TS#	NUMBER	The tablespace number
RFILE#	NUMBER	The relative file number in tablespace
STATUS	VARCHAR2(7)	The status of the file (OFFLINE|ONLINE)
ENABLED	VARCHAR2(10)	Enabled for read and/or write

Column	Datatype	Description
BYTES	NUMBER	The size of the file in bytes (from file header)
BLOCKS	NUMBER	The size of the file in blocks (from file header)
CREATE_BYTES	NUMBER	The creation size of the file (in bytes)
BLOCK_SIZE	NUMBER	The block size for the file
NAME	VARCHAR2(513)	The name of the file

V$TEMPORARY_LOBS

This view displays temporary lobs.

Column	Datatype	Description
SID	NUMBER	Session ID
CACHE_LOBS	NUMBER	Number cache temp lobs
NOCACHE_LOBS	NUMBER	Number of nocache temp 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 only for compatibility reasons. The value of this column is always equal to the value in USERNAME.
SESSION_ADDR	RAW(4)	Address of shared SQL cursor
SESSION_NUM	NUMBER	Serial number of session
SQLADDR	RAW(4)	Address of SQL statement
SQLHASH	NUMBER	Hash value of SQL statement
TABLESPACE	VARCHAR2(31)	Tablespace in which space is allocated
CONTENTS	VARCHAR2(9)	Indicates whether tablespace is TEMPORARY or PERMANENT

Column	Datatype	Description
SEGTYPE	VARCHAR2(9)	Type of sort segment: ■ 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
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
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
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is true; 0 if false

Column	Datatype	Description
MAXIORTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single read, 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	Thread status (OPEN | CLOSED)
ENABLED	VARCHAR2	Enabled status: DISABLED, (enabled) PRIVATE, or (enabled) PUBLIC
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
GROUPS	NUMBER	Number of log groups assigned to this thread
INSTANCE	VARCHAR2	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

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

This view lists valid time zone names.

Column	Datatype	Description
TZNAME	VARCHAR2(64)	The time zone region (for example, US/Pacific)
TZABBREV	VARCHAR2(64)	The corresponding daylight abbreviation (for example, PDT)

V$TRANSACTION

This view lists the active transactions in the system.

Column	Datatype	Description
ADDR	RAW(4)	Address of 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)
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)	User session object address
FLAG	NUMBER	Flag
SPACE	VARCHAR2(3)	YES if a space transaction
RECURSIVE	VARCHAR2(3)	YES if a recursive transaction

Column	Datatype	Description
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	Dependent SCN base
DSCN-W	NUMBER	Dependent SCN 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

V$TRANSACTION_ENQUEUE

V$TRANSACTION_ENQUEUE displays locks owned by transaction state objects.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	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)

Column	Datatype	Description
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 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: ■ 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$TYPE_SIZE

This view lists the sizes of various database components for use in estimating data block capacity.

Column	Datatype	Description
COMPONENT	VARCHAR2	Component name, such as segment or buffer header
TYPE	VARCHAR2	Component type
DESCRIPTION	VARCHAR2	Description of component
TYPE_SIZE	NUMBER	Size of 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. This view is available in both automatic undo management mode and 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
BEGIN_TIME	DATE	Identifies the beginning of the time interval
END_TIME	DATE	Identifies the end of the time interval
UNDOTSN	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.
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
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.
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.

Column	Datatype	Description
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.

V$VERSION

Version numbers of core library components in the Oracle server. There is one row for each component.

Column	Datatype	Description
BANNER	VARCHAR2	Component name and version number

V$VPD_POLICY

This view lists all the fine-grained security policies and predicates associated with the cursors currently in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4)	The cursor address
PARADDR	RAW(4)	The parent cursor address
SQL_HASH	NUMBER	SQL hash number
CHILD_NUMBER	NUMBER	The 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(4096)	Predicate for the policy, truncated to 4096 bytes in length

V$WAITSTAT

This view lists block contention statistics. This table is only updated when timed statistics are enabled.

Column	Datatype	Description
CLASS	VARCHAR2	Class of 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

4

Database Limits

This chapter 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

See Also: *Oracle9i SQL Reference* for information on the rules and limitations governing the use of datatypes, naming database objects and parts of 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	2 GB, regardless of the maximum file size allowed by the operating system
Database file size	Maximum	Operating system dependent. Limited by maximum operating system file size; typically 2^{22} or 4M blocks
MAXEXTENTS	Default value	Derived from tablespace default storage or DB_BLOCK_SIZE initialization parameter
	Maximum	Unlimited

Item	Type of Limit	Limit Value
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 K
		Number of tablespaces cannot exceed the number of database files, as each tablespace must include at least one file

Logical Database Limits

Item	Type	Limit
GROUP BY clause	Maximum length	The GROUP BY expression and all of the nondistinct aggregates 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

Item	Type	Limit
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 K-1 partitions
Rollback Segments	Maximum number per database	No limit; limited within a session by the MAX_ROLLBACK_SEGMENTS initialization parameter
Rows	Maximum number per table	Unlimited
SQL Statement Length	Maximum length of statements	64 K maximum; particular tools may impose lower limits
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

Item	Type	Limit
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 K; 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

5

SQL Scripts

This chapter describes SQL scripts that are required for optimal operation of Oracle.

The SQL scripts are described in the following sections:

- Creating the Data Dictionary
- Creating Additional Data Dictionary Structures
- The "NO" Scripts
- Database Migration 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 5–1 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 5–1 Creating the Data Dictionary Scripts

Script Name	Needed For	Description
`catalog.sql`	All databases	Creates the data dictionary and public synonyms for many of its views Grants `PUBLIC` access to the synonyms
`catproc.sql`	All databases	Runs all scripts required for, or used with PL/SQL
`catclust.sql`	Real Application Clusters	Creates Oracle9*i* 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
- *Oracle9i Database Administrator's Guide* for more information about creating a database
- Your release notes and *Oracle9i Database Migration* 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 5–2.

> **See Also:** Your operating system-specific Oracle documentation for the exact names and locations of these scripts on your operating system

Table 5–2 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 Oracle9*i* 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_OWNER` role	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

Table 5–2 (Cont.) Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
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
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 Oracle9*i*. 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 Oracle9*i*. 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.

Table 5–2 (Cont.) Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
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
utlscln.sql	Oracle Replication	Any user	Copies a snapshot schema from another snapshot site
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 5–3 are used to remove dictionary information for various optional services or components.

> **See Also:** *Oracle9i Database Migration* for more information about these scripts

Table 5–3 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

Database Migration Scripts

The scripts in Table 5–4 are useful 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 5–4 Database Migration Scripts

Script Name	Needed For	Description
`cmpdbmig.SQL`	Upgrading components	Runs the database migration scripts for all components that can be upgraded when connected as a user with SYSDBA privileges
`d0801070.sql`	Downgrading to release 8.1.7 of Oracle	Provides a direct downgrade path from your current release of Oracle to release 8.1.7
`d0900010.sql`	Downgrading to release 9.0.1 of Oracle	Provides a direct downgrade path from your current release of Oracle to release 9.0.1
`u0703040.sql`	Upgrading from release 7.3.4 of Oracle	Provides a direct upgrade path from release 7.3.4 of Oracle to the new release
`u0800060.sql`	Upgrading from release 8.0.6 of Oracle	Provides a direct upgrade path from release 8.0.6 of Oracle to the new release
`u0801070.sql`	Upgrading from release 8.1.7 of Oracle	Provides a direct upgrade path from release 8.1.7 of Oracle to the new release
`u0900010.sql`	Upgrading from release 9.0.1 of Oracle	Provides a direct upgrade path from release 9.0.1 of Oracle to the new release
`utlincmp.sql`	Removing incompatibilities before a downgrade	Contains a set of queries that identify database objects that are incompatible with previous releases of Oracle

See Also: *Oracle9i Database Migration* for more information about these scripts

Java Scripts

The scripts in Table 5–5 are useful only if the JServer option is installed.

See Also: *Oracle9i Java Developer's Guide*

Table 5–5 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

A

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.

Topics covered include:

- 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$SESSION_EVENT" on page 3-127, "V$SESSION_WAIT" on page 3-130, and "V$SYSTEM_EVENT" on page 3-165

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 events:

```
SELECT * FROM V$EVENT_NAME;
```

Wait Events and Parameters

The following wait events are present in the Oracle database server. 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 on ATTACH			
ARCH wait on DETACH			
ARCH wait on SENDREQ			
async disk IO	count	intr	timeout
background parameter adjustment	parameter number		
BFILE check if exists			

Event Name	P1	P2	P3
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 global cache	file#	block#	id
buffer busy global CR	file#	block#	id
buffer busy waits	file#	block#	id
buffer deadlock	dba	class*10+mode	flag
buffer for checkpoint	buffer#	dba	state*10+mode
buffer latch	latch addr	chain#	not used
buffer read retry	file#	block#	not used
checkpoint completed	not used	not used	not used
checkpoint range buffer not saved	not used	not used	not used
Contacting SCN server or SCN lock master			
control file parallel write	files	blocks	requests
control file sequential read	file#	block#	blocks
control file single write	file#	block#	blocks
conversion file read	file#	block#	blocks
db file parallel read	files	blocks	requests
db file parallel write	files	blocks	requests
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 write	file number	first dba	block cnt
dispatcher shutdown			

Event Name	P1	P2	P3
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
file identify	fib	file name	opcode
file open	fib	iov	0
free buffer waits	file#	block#	set-id#
free global transaction table entry	tries	not used	not used
free process state object	not used	not used	not used
global cache bg acks			
global cache cr request			
global cache freelist wait	lenum	not used	not used
global cache lock busy	file#	block#	lenum
global cache lock cleanup	file#	block#	lenum
global cache lock null to s	file#	block#	lenum
global cache lock null to x	file#	block#	lenum
global cache lock open null	file#	block#	class
global cache lock open s	file#	block#	lenum
global cache lock open x	file#	block#	lenum
global cache lock s to x	file#	block#	lenum
global cache multiple locks			
global cache pending ast			
imm op	msg ptr	not used	not used
inactive session	session#	waited	not used
inactive transaction branch	branch#	waited	not used
index block split	rootdba	level	childdba
instance recovery	undo segment#	not used	not used
instance state change	layer	value	waited
io done	msg ptr		
jobq slave wait			
KJC Snd: Queue / Send client message			
KJC: Wait for msg sends to complete			

Event Name	P1	P2	P3
KJCTS client waiting for tickets			
kksfbc child completion			
kksfbc research			
kkshgnc reloop			
kksscl hash split			
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	
lmon waiting for lmd/lms to freeze in step 14			
lmon waiting for lmd/lms to unfreeze in step 14			
LNS wait on ATTACH			
LNS wait on DETACH			
LNS wait on LGWR			
LNS wait on SENDREQ			
local write wait	file#	block#	
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)			

Event Name	P1	P2	P3
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		
pipe get	handle address	buffer length	timeout
pipe put	handle address	record length	timeout
PL / SQL lock timer	duration		
pmon timer	duration		
process startup	type	process#	waited
PX create server	nservers	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: 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	
PX Deq: Parse Reply	sleeptime / senderid	passes	

Event Name	P1	P2	P3
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			

Event Name	P1	P2	P3
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			
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			
sbtpcbackup			
sbtpccancel			
sbtpccommit			
sbtpcend			
sbtpcquerybackup			
sbtpcqueryrestore			

Event Name	P1	P2	P3
sbtpcrestore			
sbtpcstart			
sbtpcstatus			
sbtpcvalidate			
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
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			
switch logfile command			

Event Name	P1	P2	P3
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 EMON to spawn			
wait for Logical Standby Apply shutdown			
wait for membership synchronization			
wait for message ack			
wait for name service busy			
wait for possible quiesce finish			
wait for record update			
wait for rr lock release			
wait for stopper event to be increased			
wait for sync ack	cluinc	pending_nd	
wait for tmc2 to complete			
wait for unread message on broadcast channel	channel context	channel handle	

Event Name	P1	P2	P3
wait for unread message on multiple broadcast channels	channel context	channel handle count	
wait for verification ack	cluinc	pending_insts	
wait for votes			
waiting for slave message			
waiting to get 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

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).

lenum

The relative index number into V$LOCK_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 A–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

Parameters:

waited 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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

BFILE check if open

The session waits to check if an external large object (LOB) has already been opened.

Wait Time: The total elapsed time for the **isopen** call

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

BFILE closure

The session waits for an external large object (LOB) to close.

Wait Time: The total elapsed time for the **close** call

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

BFILE open

The session waits to check if an external large object (LOB) has already been opened.

Wait Time: The total elapsed time for the **isopen** call

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

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.

Parameters:

file#	See "file#" on page A-12
block#	See "block#" on page A-11
id	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.

Parameters:

class	See "class" on page A-12
mode	See "mode" on page A-12
flag	The flag points to the internal flags used by the session to get this block
dba	See "dba" on page A-12

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

Parameters:

dba	See "dba" on page A-12
state	State refers to the status of the buffer contents
mode	See "mode" on page A-12
buffer#	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

Parameters:

latch addr	The virtual address in the SGA where this latch is located. Use the following statement to find the name of this latch:

```
select *
from v$latch a, v$latchname b
where addr = latch addr
and a.latch# = b.latch#;
```

chain# 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

Parameters:

file# See "file#" on page A-12

block# See "block#" on page A-11

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

Parameters:

files	The number of control files to which the session is writing
blocks	The number of blocks that the session is writing to the control file
requests	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

Parameters:

file#	The controlfile from which the session is reading
block#	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).
blocks	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

Parameters:

file#	This identifies the control file to which the session is currently writing
block#	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).
blocks	The number of blocks that the session is trying to read

conversion file read

This event occurs during a 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

Parameters:

file#	The control file to which the session is currently writing
block#	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).
blocks	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

Parameters:

files	This indicates the number of files to which the session is reading

blocks	This indicates the total number of blocks to be read
requests	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

Parameters:

files	This indicates the number of files to which the session is writing
blocks	This indicates the total number of blocks to be written
requests	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

Parameters:

file#	See "file#" on page A-12
block#	See "block#" on page A-12
blocks	The number of blocks that the session is trying to read from the *file#* starting at *block#*

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

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
blocks	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

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
blocks	This is the number of blocks that the session is trying to write in *file#* starting at *block#*

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 B, "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:

file	See "file#" on page A-12

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.

Parameters:

name	See "name and type" on page A-13
mode	See "mode" on page A-12
id1	See "id1" on page A-12
id2	See "id2" on page A-12

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.

Parameters:

descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	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.

Parameters:

descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	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:

waited	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:

sleep time	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:

dba	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 B, "Oracle Enqueue Names").

Wait Time: Depends on the enqueue name

Parameters:

name	See "name and type" on page A-13
mode	See "mode" on page A-12

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 C-7.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12
block#	See "block#" on page A-11

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:

tries	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, 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 freelist wait

All releasable locks are used and a new one has been requested. To make a lock element available, a lock element is pinged.

Wait Time: The duration of the lock get operation to ping the lock element

Parameter:

lenum	See "lenum" on page A-8

global cache lock busy

The session waits to convert a buffer up from Shared Current to Exclusive Current status.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
lenum	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

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
lenum	See "lenum" on page A-8

global cache lock null to s

The session waits for a lock convert from NULL to SHARED mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12

block#	See"block#" on page A-12
class	See "class" on page A-12

global cache lock null to x

The session waits for a lock convert from NULL to EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
lenum	See"lenum" on page A-12

global cache lock open null

The session waits for a lock get in NULL mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12
class	See"class" on page A-12

global cache lock open s

The session waits for a lock get in SHARED mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-12
block#	See"block#" on page A-12

class See"class" on page A-12

global cache lock open x

The session waits for a lock get in EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file# See "file#" on page A-12

block# See"block#" on page A-12

lenum See"lenum" on page A-12

global cache lock s to x

The session waits for a lock convert from SHARED to EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file# See "file#" on page A-12

block# See"block#" on page A-12

lenum See"lenum" on page A-12

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

Parameters:

session#	See "session#" on page A-14
waited	See "waited" on page A-14

inactive transaction branch

The session waits for a transaction branch that is currently used by another session.

Wait Time: 1 second

Parameters:

branch#	The serial number of the transaction for which the session is waiting
waited	See "waited" on page A-14

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

Parameters:

rootdba	The root of the index
level	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).
childdba	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:

undo segment# If the value is 0, SMON is probably performing instance recovery. If P1 > 0, use this query to find the undo segment:

```
select *
from v$rollstat
where usn = undo segment#;
```

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)

Parameters:

layer 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.

value This value can be 0 (disable) or 1 (enable)

waited 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:

msg ptr 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

Parameters:

count	The number of LCK processes that have finished
loops	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

Parameters:

address	The address of the latch that is being checked

number	The latch number of the latch that has activity. To find more information on the latch, use the following SQL statement:
	`select *`
	`from v$latchname`
	`where latch# = number;`

process#	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.

Parameters:

address	The address of the latch for which the process is waiting

number The latch number that indexes in the V$LATCHNAME view.To find more information on the latch, use the following SQL statement:

```
select *
from v$latchname
where latch# = number;
```

tries 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)

Parameters:

object address Address of the object being loaded

lock address Address of load lock being used

mask 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)

Parameters:

handle address Address of the object being loaded

lock address Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is a State Object.

mode	Indicates the data pieces of the object which need to be loaded
namespace	See "namespace" on page A-13

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)

Parameters:

handle address	Address of the object being loaded
pin address	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.
mode	Indicates which data pieces of the object that needs to be loaded
namespace	See "namespace" on page A-13

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:

waittime	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.

Parameters:

files	Number of files to be written
blocks	Number of blocks to be written
requests	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)

Parameters:

log#	The relative sequence number of the logfiles within a log group (used only when dumping the logfiles)
block#	See "block#" on page A-11
blocks	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

Parameters:

log#	This is the number of the group/log to which the session is currently writing
block#	See "block#" on page A-11
blocks	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:

buffer#	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:

thread#	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

parallel execution create server

Used when creating or starting a parallel execution slave.

Wait Time: The time it takes to start all of the requested parallel execution slaves

Parameters:

nservers	The number of parallel execution slaves that are being started
sleeptime	Time it takes to get the processes started. The process should be started within *sleeptime*
enqueue	The number of blocks to read

parallel execution 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.

Parameters:

reason	The reason for dequeuing
sleeptime	The amount of time that the session slept
loop	The total number of times that the session has slept

parallel execution 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

Parameters:

function	Indicates the type of wait that the session is doing
sleeptime	The amount of time that the session waits (in hundredths of a second)
qref	The address of the process queue for which the session is waits

parallel execution 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

Parameters:

nalive	The number of parallel execution slaves that are still running
sleeptime	The total sleeptime since the session started to wait on this event
loop	The number of times the session waited for this event

parallel execution 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

Parameters:

serial	The serial number of the slave process queue
error	The error that has occurred
nbusy	The number of slave processes that are still busy

pending global transaction(s)

This event should happen only during testing. The session waits for pending transactions to clear.

Wait Time: 30 seconds

Parameter:

scans	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

Parameters:

handle address	The library cache object handle for this pipe
buffer length	The length of the buffer
timeout	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

Parameters:

handle address	The library cache object handle for this pipe

record length	The length of the record or buffer that has been put into the pipe
timeout	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:

duration	The duration that the user specified in the DBMS_LOCK.SLEEP or USER_LOCK.SLEEP procedures

pmon rdomain attach

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

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:

duration	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.

Parameters:

type	The process type that was started
process#	The process number of the process being started
waited	Cumulative time waited for the process to start

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*

Parameters:

queue id	The ID of the OLTP queue for which this session is waiting
process#	The process number of the process in which this session runs
wait time	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:

timeout	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*.

Parameters:

from_process	The background process for which the session is waiting. The wait is for a reply to an IPC message sent by the session.
timeout	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.

Parameters:

cache id	The CACHE# column value in the V$ROWCACHE view
mode	See "mode" on page A-12
request	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)

Parameters:

sleeptime	The amount of time that SMON tries to wait on this event in seconds
failed	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

Parameters:

driver id	See "driver id" on page A-12
break?	See "break?" on page A-12

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

Parameters:

driver id	See "driver id" on page A-12
break?	See "break?" on page A-12

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

Parameters:

driver id	See "driver id" on page A-12
#bytes	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.

Parameters:

driver id	See "driver id" on page A-12
#bytes	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

Parameters:

driver id	See "driver id" on page A-12
#bytes	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

Parameters:

driver id	See "driver id" on page A-12
#bytes	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)

Parameters:

driver id	See "driver id" on page A-12
#bytes	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)

Parameters:

driver id	See "driver id" on page A-12
#bytes	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

Parameters:

driver id	See "driver id" on page A-12

#bytes	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

Parameters:

driver id	See "driver id" on page A-12
#bytes	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

Parameters:

undo seg#	The rollback segment ID
slot#	The slot ID inside the rollback segment
wrap#	The sequence number that is incremented for each transaction

count	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:

segment#	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

Parameters:

segment#	The ID of the rollback segment that contains the transaction that is being rolled back
tx flags	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

Parameters:

segment#	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

Parameters:

circuit#	Indicates the virtual circuit# being waited on
status	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

Parameters:

file#	The rollback segment id that contains the transaction that is being rolled back
block#	The transaction flags (options) set for the transaction that is being rolled back
id	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

Parameters:

bythread#	The rollback segment id that contains the transaction that is being rolled back
ourthread#	The current instance thread number

B

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:

- *Oracle9i Application Developer's Guide - Advanced Queuing* for more information on enqueues
- "DBA_LOCK_INTERNAL" on page 2-173 and "DBA_LOCK" on page 2-172

The Oracle enqueues are:

- BL, Buffer Cache Management
- CF, Controlfile Transaction
- CI, Cross-instance Call Invocation
- CU, Bind Enqueue
- DF, Datafile

- DL, Direct Loader Index Creation
- DM, Database Mount
- DR, Distributed Recovery
- DX, Distributed TX
- FS, File Set
- IN, Instance Number
- IR, Instance Recovery
- IS, Instance State
- IV, Library Cache Invalidation
- JQ, Job Queue
- KK, Redo Log "Kick"
- L[A-P], Library Cache Lock
- MR, Media Recovery
- N[A-Z], Library Cache Pin
- PF, Password File
- PI, Parallel Slaves
- PR, Process Startup
- PS, Parallel Slave Synchronization
- Q[A-Z], Row Cache
- RT, Redo Thread
- SC, System Commit Number
- SM, SMON
- SQ, Sequence Number Enqueue
- SR, Synchronized Replication
- SS, Sort Segment
- ST, Space Management Transaction
- SV, Sequence Number Value
- TA, Transaction Recovery

- TM, DML 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

C

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.

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$SESSTAT" on page 3-131, "V$STATNAME" on page 3-159, and "V$SYSSTAT" on page 3-164 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 C–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: ■ Individual file checkpoints for operations such as offline or begin backup ■ Foreground (user-requested) checkpoints (for example, performed by `ALTER SYSTEM CHECKPOINT LOCAL` statements)	
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.	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
bytes received via SQL*Net from client	1	Total number of bytes received from the client over Net8	
bytes received via SQL*Net from dblink	1	Total number of bytes received from a database link over Net8	
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.	
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 kcsgrs 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 the `TIME_STATISTICS` parameter 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`	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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.	
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

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	Oracle Real Application Cluster only: Number of blocks written to satisfy a lock request from another instance	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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.	
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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.	
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	Oracle9*i* Real Application Clusters only: Number of blocks that encountered a corruption or checksum failure during interconnect	
global cache convert time	40	Oracle9*i* 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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	
messages sent	128		
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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)	
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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		
Parallel operations downgraded 50 to 75 pct	32		
Parallel operations downgraded 75 to 99 pct	32		
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 milliseconds	3
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.	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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	
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

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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 an 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.	
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 LWGR 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"	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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.	
sorts (disk)	64	Number of sort operations that required at least one disk write Sorts that require I/O to disk are quite resource intensive. Try increasing the size of the initialization parameter SORT_AREA_SIZE. For more information, see "SORT_ AREA_SIZE" on page 1-153.	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
sorts (memory)	64	Number of sort operations that were performed completely in memory and did not require any disk writes 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.	
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	Number of rows that are fetched using a ROWID (usually recovered from an index) 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.	
table fetch continued row	64	Number of times a chained or migrated row is encountered during a fetch Retrieving rows that span more than one block increases the logical I/O by a factor that corresponds to the number of blocks than need to be accessed. Exporting and re-importing may eliminate this problem. Evaluate the settings for the storage parameters PCTFREE and PCTUSED. This problem cannot be fixed if rows are larger than database blocks (for example, if the `LONG` datatype is used and the rows are extremely large).	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
table scan blocks gotten	64	During scanning operations, each row is retrieved sequentially by Oracle. This statistic counts the number of blocks encountered during the scan. 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.	
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 CACHE option enabled	
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	

Table C–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
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.	
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 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		

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W