Oracle8*i*

Reference

Release 8.1.5

February, 1999

Part No. A67790-01

Oracle 8i Reference, Release 8.1.5

Part No. A67790-01

Primary Author: Jason Durbin

Contributors: Gaurav Bhatia, John Bellemore, Roger Bodamer, Bill Bridge, Allen Brumm, Sumanta Chatterjee, Greg Doherty, Alan Downing, Masane Hayashi, Alex Ho, Amit Jasuja, Robert Jenkins, Boris Klots, Anjo Kolk, Susan Kotsovolos, Andre Kruglikov, Michael Kung, Paul Lane, Gordon Larimer, Nina Lewis, Juan Loaiza, William Maimone, Reza Monajjemi, Gary Ngai, Anil Nori, Alok Pareek, Kant Patel, Thomas Pfaeffle, Tuomas Pystynen, Mark Ramacher, Sriram Samu, Richard Sarwal, Ekrem Soylemez, Anand Srinivasan, James Stamos, Alvin To, Alex Tsukerman, Radek Vingralek, William Waddington, Joyo Wijaya, Graham Wood, Gail Yamanaka, Sofia Yeung

Graphic Designer: Valarie Moore

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

Contents

2 Static Data Dictionary Views

A Oracle Wait Events

B Enqueue and Lock Names

C Statistics Descriptions

Send Us Your Comments

Oracle8i Reference, Release 8.1.5

Part No. A67790-01

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

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

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

- email - infodev@us.oracle.com
- FAX - (650)506-7228
- Street Address:
  Oracle Corporation
  Oracle Server Documentation Manager
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

Preface

This manual provides reference information about Oracle8*i* for all operating systems.

This information includes:

- Initialization Parameters
- Static Data Dictionary Views
- Dynamic Performance (V$) Views
- Database Limits
- SQL Scripts
- Oracle Wait Events
- Enqueue and Lock Names
- Statistics Descriptions

Feature Coverage and Availability

Oracle8i Reference contains information relating to both Oracle8*i* and the Oracle8*i* Enterprise Edition. Some of the features documented in this manual are available only if you have purchased the Oracle8*i* Enterprise Edition.

For information about the differences between Oracle8*i* and the Oracle8*i* Enterprise Edition, please refer to *Getting to Know Oracle8i and the Oracle8i Enterprise Edition*. *Oracle8i Reference* describes those features which are common to both products, and those which are only available with the Oracle8*i* Enterprise Edition or a particular option.

Audience

This manual is intended for database administrators, system administrators, and database application developers.

Knowledge Assumed of the Reader

It is assumed that readers of this manual are familiar with relational database concepts, basic Oracle concepts, and with the operating system environment under which they are running Oracle.

Installation and Migration Information

This manual is *not* an installation or migration guide. If your primary interest is installation, refer to your operating system-specific Oracle documentation. If your primary interest is database and application migration, refer to *Oracle8i Migration*.

Information Database Administration

While this manual describes the architecture, processes, structures, and other concepts of Oracle8*i*, it does not explain how to administer an Oracle server. For that information, see the *Oracle8i Administrator's Guide*.

Application Design Information

In addition to administrators, experienced users of Oracle and advanced database application designers will find information in this manual useful. However, database application developers should also refer to the *Oracle8i Application Developer's Guide - Fundamentals* and to the documentation for the tool or language product they are using to develop Oracle database applications.

How Oracle8i Reference Is Organized

This manual is organized as follows:

Chapter 1, "Initialization Parameters"

This chapter contains descriptions of the database initialization parameters in the parameter file that are required to start an instance.

Chapter 2, "Static Data Dictionary Views"

This chapter contains descriptions of the Oracle data dictionary tables and views.

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

This chapter contains descriptions of 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 server.

Appendix A, "Oracle Wait Events"

This appendix describes certain event name, wait time and parameters for wait events displayed by the V$SESSION_WAIT and V$SYSTEM_EVENT views.

Appendix B, "Enqueue and Lock Names"

This appendix describes certain enqueues and locks used by Oracle8*i*.

Appendix C, "Statistics Descriptions"

This appendix describes certain statistics stored in the V$SESSION_WAIT and V$SYSSTAT dynamic performance table.

Conventions Used in This Manual

The following sections describe the conventions used in this manual.

Text of the Manual

The text of this manual uses the following conventions.

UPPERCASE Characters

Uppercase text is used to call attention to command keywords, database object names, parameters, filenames, and so on.

For example, "After inserting the default value, Oracle checks the FOREIGN KEY integrity constraint defined on the DEPTNO column," or "If you create a private rollback segment, the name must be included in the ROLLBACK_SEGMENTS initialization parameter."

Italicized Characters

Italicized words within text are book titles or emphasized words.

Code Examples

Commands or statements of SQL, Oracle Enterprise Manager line mode (Server Manager), and SQL*Plus appear in a monospaced font.

For example:

```
INSERT INTO emp (empno, ename) VALUES (1000, ’SMITH’);
ALTER TABLESPACE users ADD DATAFILE ’users2.ora’ SIZE 50K;
```

Example statements may include punctuation, such as commas or quotation marks. All punctuation in example statements is required. All example statements terminate with a semicolon (;). Depending on the application, a semicolon or other terminator may or may not be required to end a statement.

`UPPERCASE` in Code Examples

Uppercase words in example statements indicate the keywords within Oracle SQL. When you issue statements, however, keywords are not case sensitive.

`lowercase` in Code Examples

Lowercase words in example statements indicate words supplied only for the context of the example. For example, lowercase words may indicate the name of a table, column, or file.

Your Comments Are Welcome

We value and appreciate your comment as an Oracle user and reader of our manuals. As we write, revise, and evaluate our documentation, your opinions are the most important feedback we receive.

You can send comments and suggestions about this manual to the Information Development department at the following e-mail address:

infodev@us.oracle.com

If you prefer, you can send letters or faxes containing your comments to:

Server Technologies Documentation Manager

Oracle Corporation

500 Oracle Parkway Redwood Shores, CA 94065

Fax: (650) 506-7228 Attn.: Oracle8*i* Reference

1

Initialization Parameters

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

- Initialization Parameter File
- Specifying Values in the Parameter File
- Reading the Parameter Descriptions
- Parameter Descriptions

Initialization Parameter File

The initialization parameter file is a text file that contains a list of parameters and a value for each parameter. The file should be written in the client's default character set. Specify values in the parameter file which reflect your installation.

The following are sample entries in a parameter file:

```
PROCESSES = 100
OPEN_LINKS = 12
GLOBAL_NAMES = TRUE
```

The name of the 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 on the name INIT.ORA. As the database administrator, you can choose a different filename for your parameter file. There is also an INITDW.ORA file which contains suggested parameter settings for data warehouses and data marts.

See your Oracle operating system-specific documentation for the default locations and filenames for these parameter files. The INIT.ORA file is read by the client-side tool which is being used to start the server (e.g., SQL*Plus).

Sample parameter files are provided on the Oracle Server distribution medium for each operating system. A distributed sample file is sufficient for initial use, but you will want to make changes in the file to tune the database system for best performance. Any changes will take effect after completely shutting down the instance and then restarting it.

Database administrators can use initialization parameters to do the following:

- optimize performance by adjusting memory structures, for example, the number of database buffers in memory
- set some database-wide defaults, for example, how much space is initially allocated for a context area when it is created
- set database limits, for example, the maximum number of database users
- specify names of files

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

Note: If you are using Trusted Oracle, see the Trusted Oracle documentation information for more details.

Specifying Values in the Parameter File

This section describes several aspects of setting parameter values in the parameter file. The following topics are included:

- Rules
- Using Special Characters in Parameter Values
- Changing Parameter Values
- Displaying Current Parameter Values
- Uses of Parameters
- Types of Parameters
- Parameters You Should Not Specify in the Parameter File
- When Parameters Are Set Incorrectly

Rules

The following rules govern the specification of parameters in the parameter file:

- All parameters are optional.
- Only parameters and comments should appear in the parameter file.
- A pound sign (#) starts a comment; the rest of the line is ignored.
- The server has a default value for each parameter. This value may be operating system-dependent, depending on the parameter.
- Parameters can be specified in any order.
- Case (upper or lower) in filenames is only significant 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:

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

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

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

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

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

Either syntax is valid.

Note: If you choose to enter values over several lines, make sure the parameters are contiguous. If they are not, the first line will not be processed properly. For example, in:

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

SEG1 and SEG2 will not be set correctly.

- A backslash or *escape character* (\) indicates continuation of the parameter specification. If a backslash continues a line, the continued line must have no preceding spaces. For example:

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

- The keyword IFILE can be used to call another parameter file, which must be in the same format as the original parameter file.
- If a parameter is specified consecutively "n" number of times, it is associated with a list of "n" values. If a parameter is specified "n" number of times, but not all consecutively, then the last consecutive list of values is used.
- Enclose parameter values which contain whitespace in double quotes. For example:

  ```
  NLS_TERRITORY = "CZECH REPUBLIC"
  ```

Suggestion: Listing parameters in alphabetical order in the parameter file is a good idea. That makes it easier to find them and helps ensure that each parameter is specified only once.

See your operating system-specific Oracle documentation for more information on parameter files.

Using Special Characters in Parameter Values

If a parameter value contains a special character, then either the special character must be preceded by an escape character or the entire parameter value must be contained in double quotes. For example:

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

or

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

Table 1–1 lists the special characters.

Table 1–1   Special Characters in the Initialization Parameter File

Character	Description
#	Comment
(	Start list of values
)	End list of values
"	Start or end of quoted string
'	Start of end of quoted string
=	Separator of keyword and value(s)
,	Separator of elements
-	Precedes UNIX-style keywords
\	Escape character

Wherever a special character must be treated literally in the initialization parameter file, it must be either prefaced by the escape character or the entire string that contains the special character must be surrounded by single or double quotes.

Using the Escape Character

As described in "Rules", the escape character (\) can also signify a line continuation. If the escape character is followed by an alphanumeric character, then the escape character is treated as a normal character in the input. If it is not followed by an alphanumeric, then the escape character is treated either as an escape character or as a continuation character.

Using Quotes

Quotes can be nested in any of three ways. One method is to double the quotes in the nested string. For example:

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

Another method is to alternate single and double quotes. For example:

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

The third method is to escape the inner quotes. For example:

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

Changing Parameter Values

To change a parameter's value, edit the parameter file. The next time the instance starts, it uses the new parameter values in the updated parameter file. Note that the change does not take effect until the instance is shut down and restarted.

Dynamic Parameters

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

Use this syntax for dynamically altering the initialization parameters:

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

Whenever a dynamic parameter is modified using the ALTER SYSTEM, or ALTER SYSTEM DEFERRED command, then the command that modifies the parameter is also recorded in the alert log.

The ALTER SESSION command changes the value of the parameter specific to the session that invokes this command. The value of this parameter does not change for other sessions in the instance. The value of the initialization parameters listed in Table 1–2 can be changed with ALTER SESSION.

Table 1–2 Initialization Parameters Alterable with ALTER SESSION

DB_BLOCK_CHECKING	DB_FILE_MULTIBLOCK_READ_COUNT
GLOBAL_NAMES	HASH_AREA_SIZE
HASH_JOIN_ENABLED	HASH_MULTIBLOCK_IO_COUNT
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	OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE	OPS_ADMIN_GROUP
OPTIMIZER_INDEX_CACHING	OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_MAX_PERMUTATIONS	OPTIMIZER_MODE
OPTIMIZER_PERCENT_PARALLEL	OPTIMIZER_SEARCH_LIMIT
PARALLEL_BROADCAST_ENABLED	PARALLEL_INSTANCE_GROUP
PARALLEL_MIN_PERCENT	PARTITION_VIEW_ENABLED
PLSQL_V2_COMPATIBILITY	QUERY_REWRITE_ENABLED
QUERY_REWRITE_INTEGRITY	REMOTE_DEPENDENCIES_MODE
SESSION_CACHED_CURSORS	SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE	SORT_MULTIBLOCK_READ_COUNT
STAR_TRANSFORMATION_ENABLED	TIMED_STATISTICS

The ALTER SYSTEM command modifies the global value of the parameter until the database is shut down. The ALTER SYSTEM command does not always change the parameter value for the current session. Use the ALTER SESSION command to change the parameter value for the current session. The value of the initialization parameters listed in Table 1–3 can be changed with ALTER SYSTEM.

Table 1–3 Initialization Parameters Alterable with ALTER SYSTEM

AQ_TM_PROCESSES	BACKGROUND_DUMP_DEST
CONTROL_FILE_RECORD_KEEP_TIME	CORE_DUMP_DEST
DB_BLOCK_CHECKSUM	DB_BLOCK_MAX_DIRTY_TARGET
DB_FILE_MULTIBLOCK_READ_COUNT	FAST_START_IO_TARGET
FAST_START_PARALLEL_ROLLBACK	FIXED_DATE
FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY	GC_DEFER_TIME
GLOBAL_NAMES	HASH_MULTIBLOCK_IO_COUNT
HS_AUTOREGISTER	JOB_QUEUE_PROCESSES
LICENSE_MAX_SESSIONS	LICENSE_MAX_USERS
LICENSE_SESSIONS_WARNING	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_CHECKPOINT_INTERVAL
LOG_CHECKPOINT_TIMEOUT	MAX_DUMP_FILE_SIZE
MTS_DISPATCHERS	MTS_SERVERS
PARALLEL_ADAPTIVE_MULTI_USER	PARALLEL_INSTANCE_GROUP
PARALLEL_THREADS_PER_CPU	PLSQL_V2_COMPATIBILITY
QUERY_REWRITE_ENABLED	QUERY_REWRITE_INTEGRITY
REMOTE_DEPENDENCIES_MODE	RESOURCE_LIMIT
RESOURCE_MANAGER_PLAN	STANDBY_ARCHIVE_DEST
TIMED_OS_STATISTICS	TIMED_STATISTICS
USER_DUMP_DEST	

The ALTER SYSTEM DEFERRED command 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 initialization parameters listed in Table 1–4 can be changed with ALTER SYSTEM DEFERRED.

Table 1–4 Initialization Parameters Alterable with ALTER SYSTEM DEFERRED

DB_BLOCK_CHECKING	BACKUP_TAPE_IO_SLAVES
DB_FILE_DIRECT_IO_COUNT	OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE	SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE	SORT_MULTIBLOCK_READ_COUNT
TRANSACTION_AUDITING	

Displaying Current Parameter Values

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

```
SHOW PARAMETERS
```

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

Enter the following text string to see a display for all parameters having BLOCK in their name.:

```
SHOW PARAMETERS BLOCK
```

If you display all the parameters, you might want to use the SPOOL command to write the output to a file.

Uses of Parameters

Initialization parameters can be grouped by function in several different ways. For example, there are parameters that perform the following functions:

- set database-wide limits
- set user or process limits
- name files or directories required by a database system
- set limits on database resources
- affect performance (these are called *variable parameters*)

The set of variable parameters are of particular interest to database administrators because these parameters are used primarily for improving database performance.

Types of Parameters

The Oracle Server has the following types of initialization parameters:

- Derived Parameters
- Global Cache Parameters with the Prefix GC
- Operating System-Dependent Parameters
- Variable Parameters (these can be dynamic parameters or any of the above)
- Heterogeneous Services Parameters

Derived Parameters

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

Global Cache Parameters with the Prefix GC

Initialization parameters with the prefix GC, such as GC_DEFER_TIME, apply to systems using the Oracle Parallel Server. The prefix GC stands for Global Cache. The settings of these parameters determine how the Oracle Parallel Server coordinates multiple instances. The settings you choose have an effect on the use of certain operating system resources.

Additional Information: For more information about the Parallel Server, see *Oracle8i Parallel Server Concepts and Administration*.

See your system release bulletins or other operating system-specific Oracle documentation for platform-specific information on Parallel Server parameters.

Operating System-Dependent Parameters

For some initialization parameters, the valid values or ranges depend upon the host operating system. This is denoted in the default, or range column as operating system-dependent. 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 a system-dependent default value.

See 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 11th 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.

Heterogeneous Services Parameters

There are a number of initialization parameters specific to Heterogeneous Services which must be set using a package called DBMS_HS.

For information about specifying these parameters, see *Oracle8i Distributed Database Systems*.

Parameters You Should Not Specify in the Parameter File

The following types of parameters might never have to be specified in the parameter file:

- parameters that you never alter except when instructed to do so by Oracle Corporation to resolve a problem
- derived parameters that normally do not need altering because their values are automatically calculated by the Oracle 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.

You may see messages indicating that a parameter value is too low or too high, or that you have reached the maximum for some resource. Frequently, you can wait a short while and retry the operation when the system is not as busy. If a message occurs repeatedly, you should shut down the instance, adjust the relevant parameter, and restart the instance.

Reading the Parameter Descriptions

The parameter descriptions in this chapter follow the format shown below.

PARAMETER_NAME

Parameter type:	Whether the type is integer, boolean, string, and so on.
Parameter class:	Whether the parameter is dynamic or static. If dynamic, then it also describes whether it can be changed by an ALTER SYSTEM or ALTER SESSION statement.
Default value:	The value this parameter assumes if not explicitly specified.
Range of values:	The valid range of values that this parameter can assume, shown as a minimum and maximum value. Not applicable to all parameters.
Multiple instances:	How the values for this parameter must be specified for multiple instances in an Oracle Parallel Server. Not applicable to all parameters.
OK to change:	Notes on changing the parameter value. Not specified for all releases.

The remaining paragraphs provide a textual description of the parameter and the effects of different settings.

For more information, see references to chapters or books that contain more detailed information on this subject.

Parameter Descriptions

Descriptions of the individual initialization parameters follow in alphabetical order.

Most initialization parameter values are global (on a database-wide basis), not per user, unless otherwise specified.

For more information, see your system release bulletins or other operating system-specific Oracle documentation.

O7_DICTIONARY_ACCESSIBILITY

Parameter type:	boolean
Parameter class:	static
Default value:	TRUE
Range of values:	TRUE, FALSE

O7_DICTIONARY_ACCESSIBILITY is intended to be used for migration from Oracle7 to Oracle8*i*. O7_DICTIONARY_ACCESSIBILITY controls restrictions on SYSTEM privileges. If the parameter is set to TRUE, access to objects in SYS schema is allowed (Oracle7 behavior). If this parameter is set to FALSE, SYSTEM privileges that allow access to objects in other schema do not allow access to objects in dictionary schema.

For example, if O7_DICTIONARY_ACCESSIBILITY=FALSE, then the SELECT ANY TABLE statement will allow access to views or tables in any schema except SYS schema (for example, dictionaries could not be accessed). The system privilege, EXECUTE ANY PROCEDURE will ALLOW ACCESS on the procedures in any other schema except in SYS schema.

If 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.

For more information on this parameter and the roles mentioned above, see the *Oracle8i Administrator's Guide*.

ALLOW_PARTIAL_SN_RESULTS

Obsoleted in 8.1.3.

ALWAYS_ANTI_JOIN

Parameter type:	string
Parameter class:	static
Default value:	NESTED_LOOPS
Range of values:	NESTED_LOOPS / MERGE / HASH

ALWAYS_ANTI_JOIN sets the type of antijoin that the Oracle server uses. The system checks to verify that it is legal to perform an antijoin, and if it is, processes the subquery depending on the value of this parameter. When set to the value NESTED_LOOPS, the Oracle server uses a nested loop antijoin algorithm. When set to the value MERGE, the Oracle server uses the sort merge antijoin algorithm. When set to the value HASH, the Oracle server uses the hash antijoin algorithm to evaluate the subquery.

ALWAYS_SEMI_JOIN

Parameter type:	string
Parameter class:	static
Default value:	standard
Range of values:	NESTED_LOOPS / MERGE / HASH

ALWAYS_SEMI_JOIN sets the type of semijoin that the Oracle server uses. The system checks to verify that it is legal to perform a semijoin, and if it is, processes the subquery depending on the value of this parameter. When set to the value NESTED_LOOPS, the Oracle server uses a nested loop semijoin algorithm. When set to the value MERGE, it uses the sort merge semijoin algorithm. When set to the value HASH, it uses the hash semijoin algorithm.

AQ_TM_PROCESSES

Parameter type:	integer
Parameter class:	dynamic, scope= ALTER SYSTEM
Default value:	0
Range of values:	0-10

AQ_TM_PROCESSES specifies whether a queue monitor is created. If set to 1, then one queue monitor process is created to monitor the messages. If AQ_TM_PROCESSES is not specified or is set to 0, then the queue monitor is not created.

For more information about this parameter and Advanced Queuing, see the *Oracle8i Application Developer's Guide - Fundamentals*.

ARCH_IO_SLAVES

Obsoleted in 8.1.3.

AUDIT_FILE_DEST

Parameter type:	string
Parameter class:	static
Default value:	$ORACLE_HOME/RDBMS/AUDIT

AUDIT_FILE_DEST specifies the directory where auditing files are stored.

AUDIT_TRAIL

Parameter type:	string
Parameter class:	static
Default value:	NONE
Range of values:	NONE (FALSE), DB (TRUE), OS

AUDIT_TRAIL enables or disables the writing of rows to the audit trail. Audited records are not written if the value is NONE or if the parameter is not present. The OS option enables system-wide auditing and causes audited records to be written to the operating system's audit trail. The DB option enables system-wide auditing and causes audited records to be written to the database audit trail (the SYS.AUD$ table).

The values TRUE and FALSE are also supported for backward compatibility. TRUE is equivalent to DB, and FALSE is equivalent to NONE.

The SQL AUDIT statements can set auditing options regardless of the setting of this parameter.

For more information, see the *Oracle8i Administrator's Guide*.

B_TREE_BITMAP_PLANS

Obsoleted in 8.1.3.

BACKGROUND_CORE_DUMP

Parameter type:	string
Parameter class:	static
Default value:	PARTIAL
Range of values:	FULL / PARTIAL

BACKGROUND_CORE_DUMP specifies whether the SGA is dumped as part of the generated core file. When BACKGROUND_CORE_DUMP=FULL, the SGA is dumped as part of the generated core file. If BACKGROUND_CORE_DUMP=PARTIAL, then the SGA is not dumped as part of the generated core file.

BACKGROUND_DUMP_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	operating system-dependent
Range of values:	valid local pathname, directory, or disk

BACKGROUND_DUMP_DEST specifies the pathname for a directory where debugging trace files for the background processes (LGWR, DBWn, 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-wide or globally is recorded here. This file records all instance start ups and shut downs, messages to the operator console, and errors that cause trace files to be written. It also records every CREATE, ALTER, or DROP operation on a database, tablespace, or rollback segment.

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. This file grows slowly, but without limit, so the database administrator might want to delete it periodically. The file can be deleted even when the database is running.

For more information, see the *Oracle8i Administrator's Guide*. See your operating system-specific Oracle documentation for the default value.

BACKUP_DISK_IO_SLAVES

Obsoleted in 8.1.3.

BACKUP_TAPE_IO_SLAVES

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM DEFERRED
Default value:	TRUE
Range of values:	TRUE/FALSE

BACKUP_TAPE_IO_SLAVES specifies whether I/O slaves are used by the Recovery Manager to backup, copy, or restore data to tape. When BACKUP_TAPE_IO_SLAVES = TRUE, an I/O slave process is used to write to or read from a tape device. If this parameter is FALSE (the default), then I/O slaves are not used for backups; instead, the shadow process engaged in the backup will access the tape device.

Note, as a tape device can only be accessed by one process at any given time, this parameter is a boolean, that allows or disallows deployment of an I/O slave process to access a tape device.

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

In order to perform duplexed backups, this parameter needs to be enabled, otherwise an error will be signalled. Recovery Manager will configure as many slaves as needed for the number of backup copies requested whent his parameter is enabled.

BITMAP_MERGE_AREA_SIZE

Parameter type:	integer
Parameter class:	static
Default value:	1 Mb
Range of values:	system-dependent value

BITMAP_MERGE_AREA_SIZE parameter specifies the amount of memory used to merge bitmaps retrieved from a range scan of the index. The default value is 1 Mb. A larger value should improve performance because the bitmap segments must be sorted before being merged into a single bitmap. This parameter is not dynamically alterable at the session level.

BLANK_TRIMMING

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE/FALSE

BLANK_TRIMMING specifies the data assignment semantics of character datatypes. A value of TRUE allows the data assignment of a source character string/variable to a destination character column/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 is in compliance with SQL92 Transitional Level and above semantics.

A value of 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
Parameter class:	static
Default value:	none

BUFFER_POOL_KEEP is used to improve buffer cache performance. It allows you to keep an object in the buffer cache.

For more information, see *Oracle8i Tuning*.

BUFFER_POOL_RECYCLE

Parameter type:	string
Parameter class:	static
Default value:	none

BUFFER_POOL_RECYCLE is used to improve buffer cache performance. It allows you to limit the size of an object in the buffer cache.

For more information, see *Oracle8i Tuning*.

CACHE_SIZE_THRESHOLD

Obsoleted in 8.1.3.

CCF_IO_SIZE

Obsoleted in 8.0.4. Use DB_FILE_DIRECT_IO_COUNT instead.

CLEANUP_ROLLBACK_ENTRIES

Obsoleted in 8.1.3.

CLOSE_CACHED_OPEN_CURSORS

Obsoleted in 8.1.3.

COMMIT_POINT_STRENGTH

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent
Range of values:	0 - 255

COMMIT_POINT_STRENGTH 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. A database's commit point strength should be set relative to 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, 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.

For more information about two-phase commit, see *Oracle8i Concepts* and *Oracle8i Distributed Database Systems*. See also your operating system-specific Oracle documentation for the default value.

COMPATIBLE

Parameter type:	string
Parameter class:	static
Default value:	8.1.0
Range of values:	default release to current release
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 in case it becomes necessary to revert to the earlier release. This parameter specifies the release with which the Oracle server must maintain compatibility. Some features of the current release may be restricted.

When using the standby database and 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.

This parameter allows you to immediately take advantage of the maintenance improvements of a new release in your production systems without testing the new functionality in your environment.

The default value is the earliest release with which compatibility can be guaranteed.

The value of COMPATIBLE must be set to 8.1 or higher in order to use stored columns of UROWID type. ROWID pseudo-columns for index-organized tables will work irrespective of the parameter setting.

For more information, see *Oracle8i Migration*. See also your operating system-specific Oracle documentation for the default value.

COMPATIBLE_NO_RECOVERY

Obsoleted in 8.1.3.

COMPLEX_VIEW_MERGING

Obsoleted in 8.1.3.

CONTROL_FILE_RECORD_KEEP_TIME

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	7 (days)
Range of values:	0 - 365 (days)

Records in some sections in the control file are circularly reusable while records in other sections are never reused. CONTROL_FILE_RECORD_KEEP_TIME applies to reusable sections.

It specifies the minimum age in days that a record must have before it 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 CONTROL_FILE_RECORD_KEEP_TIME is set to 0, then reusable sections never expand and records are reused as needed.

Table 1–5 lists the names of reusable sections.

Table 1–5 Names of Reusable Sections

ARCHIVED LOG	BACKUP CORRUPTION
BACKUP DATAFILE	BACKUP PIECE
BACKUP REDO LOG	BACKUP SET
COPY CORRUPTION	DATAFILE COPY
DELETED OBJECT	LOGHISTORY
OFFLINE RANGE	

CONTROL_FILES

Parameter type:	string
Parameter class:	static
Default value:	operating system-dependent
Range of values:	1 - 8 filenames

CONTROL_FILES specifies one or more names of control files, separated by commas. Oracle Corporation recommends using multiple files on different devices or mirroring the file at the OS level.

For more information, see the *Oracle8i Administrator's Guide*.

CORE_DUMP_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	$ORACLE_HOME/DBS/

CORE_DUMP_DEST specifies the directory where core files are dumped.

CPU_COUNT

Parameter type:	integer
Parameter class:	static
Default value:	automatically set by Oracle
Range of values:	0 - unlimited
OK to change:	no

CPU_COUNT specifies the number of CPUs available to Oracle. Oracle uses it to set the default value of the LOG_SIMULTANEOUS_COPIES parameter. On single-CPU computers, the value of CPU_COUNT is 1.

Warning: 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.

If there is heavy contention for latches, change the value of LOG_ SIMULTANEOUS_COPIES to twice the number of CPUs you have. Do not change the value of CPU_COUNT.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for information about this parameter.

CREATE_BITMAP_AREA_SIZE

Parameter type:	integer
Parameter class:	static
Default value:	8 Mb
Range of values:	operating system-dependent

CREATE_BITMAP_AREA_SIZE specifies the amount of memory allocated for bitmap creation. The default value is 8 Mb. A larger value might lead to faster index creation. 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. This parameter is not dynamically alterable at the session level.

CURSOR_SPACE_FOR_TIME

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE / FALSE

If CURSOR_SPACE_FOR_TIME is set to TRUE, the database uses more space for cursors to save time. It affects both the shared SQL area and the client's private SQL area.

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

Setting this parameter to TRUE also retains the private SQL area allocated for each cursor between executes instead of discarding it after cursor execution. This saves cursor allocation and initialization time.

For more information, see *Oracle8i Concepts*.

DB_BLOCK_BUFFERS

Parameter type:	integer
Parameter class:	static
Default value:	32768 buffers
Range of values:	4 - operating system-specific
OK to change:	yes
Multiple instances:	can have different values

DB_BLOCK_BUFFERS specifies the number of database buffers available in the buffer cache. It is one of the primary parameters which contribute to the total memory requirements of the SGA on the instance.

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

This parameter affects the probability that a data block will be pinged when Parallel Server is enabled: the more buffers, the more chance of pings.

Note: The checkpoint process is enabled by default if the system configuration is such that moving checkpoint processing from the logwriter to a separate process is deemed to improve performance. Currently, the criteria are DB_FILES >= 50 or DB_BLOCK_BUFFERS >= 10000.

For more information, see *Oracle8i Concepts*. See also your operating system-specific Oracle documentation.

DB_BLOCK_CHECKPOINT_BATCH

Obsoleted in 8.1.3.

DB_BLOCK_CHECKING

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	FALSE
Range of values:	TRUE / FALSE

DB_BLOCK_CHECKING is used to control whether block checking is done fro transaction managed blocks. As early detection of corruptions is ueful, and has minimal, if any, performance impact, Oracle recommends that the default setting be used. The FALSE setting is provided for compatibility with ealier releases where block checking is disabled as a default. As the parameter is dynamic, it provides more flexibility than events 10210 and 10211, which it will ultimately replace. Note that the setting of DB_BLOCK_CHECKING overrides any setting of events 10210 and 10211.

DB_BLOCK_CHECKSUM

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	FALSE
Range of values:	TRUE/FALSE

If DB_BLOCK_CHECKSUM is set to TRUE, DBWn and the direct loader will calculate a checksum and store it in the cache header of every data block when writing it to disk. Checksums will be verified when a block is read only if this parameter is TRUE and the last write of the block stored a checksum.

If DB_BLOCK_CHECKSUM is set to TRUE, every log block will also be given a checksum before it is written to the current log.

Warning: Setting DB_BLOCK_CHECKSUM to TRUE can cause performance overhead.

For more information, see the *Oracle8i Administrator's Guide*.

DB_BLOCK_LRU_EXTENDED_STATISTICS

Obsoleted in 8.1.3.

DB_BLOCK_LRU_LATCHES

Parameter type:	integer
Parameter class:	static
Default value:	CPU_COUNT/2
Range of values:	1 - twice the number of CPUs

DB_BLOCK_LRU_LATCHES specifies the upper bound of the number of LRU latch sets. Set this parameter to a value equal to the desired number of LRU latch sets. Oracle decides whether to use this value or reduce it based on a number of internal checks. If the parameter is not set, Oracle calculates a value for the number of sets. The value calculated by Oracle is usually adequate. Increase this only if misses are higher than 3% in V$LATCH.

DB_BLOCK_LRU_STATISTICS

Obsoleted in 8.1.3.

DB_BLOCK_MAX_DIRTY_TARGET

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM set at runtime
Default value:	all the buffers in the cache
Range of values:	1000 to all buffers in the cache, setting to 0 disables limiting dirty buffers
OK to change:	yes, at run-time
Multiple instances:	can have different values

DB_BLOCK_MAX_DIRTY_TARGET specifies the number of buffers that can be dirty (modified and different from what is on disk) in the buffer cache. It indirectly specifies a rough limit on the number of blocks that must be read during crash and instance recovery.

Note that the FAST_START_IO_TARGET parameter, (*available only with the Oracle Enterprise Edition*), can be used to enforce a a more accurate bound on the number of IOs during recovery.)

DBWn will write out buffers to attempt to limit the number of dirty buffers in the cache below the specified value.

Setting this value to 0 disables the mechanism that limits the number of dirty buffers in the cache. All other writing activity continues as before (that is, it is unaffected by setting this parameter to 0).

DB_BLOCK_SIZE

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent
Range of values:	operating system-dependent (2048 - 32768)
OK to change:	only at database creation
Multiple instances:	must have the same value

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 CREATE DATABASE time determines the size of the blocks; at all other times the value must be set to the original value.

This parameter affects the maximum value of the FREELISTS storage parameter for tables and indexes. DSS (data warehouse) database environments tend to benefit from larger block size values.

For more information about block size, see *Oracle8i Concepts*. See also your operating system-specific Oracle documentation for the default value.

DB_DOMAIN

Parameter type:	string
Parameter class:	static
Default value:	WORLD
Range of values:	any legal string of name components, separated by periods and up to 128 characters long, including periods (see valid characters below) - this value cannot be NULL
Multiple instances:	must have the same value

DB_DOMAIN specifies the extension components of a global database name, consisting of valid identifiers, separated by periods. Specifying DB_DOMAIN as a unique string for all databases in a domain is recommended. For example, 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 = "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. See also the view GLOBAL_NAME.

The following characters are valid in a database domain name:

- alphabetic characters
- numbers
- underscore (_)
- pound (#)

For more information, see the *Oracle8i Administrator's Guide*.

DB_FILE_DIRECT_IO_COUNT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM DEFERRED
Default value:	64
Range of values:	operating system-dependent

DB_FILE_DIRECT_IO_COUNT is used to specify the number of blocks to be used for I/O operations done by backup, restore or direct path read and write functions. The I/O buffer size is a product of DB_FILE_DIRECT_IO_COUNT and DB_BLOCK_SIZE. The I/O buffer size cannot exceed max_IO_size for your platform.

Assigning a high value to this parameter results in greater use of PGA or SGA memory.

Note: If you have previously used CCF_IO_SIZE and are migrating to DB_FILE_DIRECT_IO_COUNT, remember that CCF_IO_SIZE was specified in bytes while DB_FILE_DIRECT_IO_COUNT must be specified in database blocks.

DB_FILE_MULTIBLOCK_READ_COUNT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	8
Range of values:	operating system-dependent

DB_FILE_MULTIBLOCK_READ_COUNT is used for multi-block I/O and 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 factors such as these:

- the size of the table
- the multi-block read count
- whether parallel execution is being utilized for the operation

The default is 8. OLTP and batch environments typically have values for this parameter in the range of 4 to 16. DSS (data warehouse) database environments tend to get the most benefit from maximizing the value for this parameter.

The actual maximums vary by operating system; they are always less than the operating system's maximum I/O size expressed as Oracle blocks (*max_IO_size*/DB_BLOCK_SIZE). Attempts to set this parameter to a value greater than the maximum will cause the maximum to be used.

For information on the optimizer, see *Oracle8i Tuning*. See also your operating system-specific Oracle documentation for the default value.

DB_FILE_NAME_CONVERT

Parameter type:	string
Parameter class:	static
Default value:	none
Range of values:	character string

Use DB_FILE_NAME_CONVERT to convert the filename of a new data file on the primary database to a filename on the standby database.

Adding a datafile to the primary database necessitates adding a corresponding file to the standby database. When the standby database is updated, this parameter is used to convert the datafile name on the primary database to the a datafile name on the standby database. The file must exist and be writable on the standby database or the recovery process will halt with an error.

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.

DB_FILE_NAME_CONVERT can also be used to rename the datafiles in the clone coltrolfile when setting up the clone database during Tablespace Point-In-Time Recovery. For more information, see Oracle8i Backup and Recovery Guide.

DB_FILE_SIMULTANEOUS_WRITES

Obsoleted in 8.1.3.

DB_FILES

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent; 200 on Solaris
Range of values:	minimum value: either the value that was specified in the MAXDATAFILES clause the last time CREATE DATABASE or CREATE CONTROLFILE was executed, or the current actual number of datafiles in the data maximum value: operating system-dependent
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 for DB_FILES 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 statement.

If you increase the value of DB_FILES, you must shut down and restart all instances accessing the database before the new value can take effect.

Note: The checkpoint process is enabled by default if the system configuration is such that moving checkpoint processing from the logwriter to a separate process is deemed to improve performance. Currently, the criteria are DB_FILES >= 50 or DB_BLOCK_BUFFERS >= 10000.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value.

DB_NAME

Parameter type:	string
Parameter class:	static
Default value:	NULL
Range of values:	any valid database name
Multiple instances:	must have the same value, or else the same value must be specified in STARTUP OPEN *db_name* or ALTER DATABASE *db_name* MOUNT

DB_NAME can specify a database identifier of up to eight characters. If specified, it must correspond to the name specified in the CREATE DATABASE statement. Although the use of DB_NAME is optional, it should generally be set before invoking CREATE DATABASE and then referenced in that statement.

If not specified, a database name must appear on either the STARTUP or the ALTER DATABASE MOUNT command line for each instance of the parallel server.

The following are valid characters in a database name:

- alphabetic characters
- numbers
- underscore (_)
- pound (#)
- dollar symbol ($)

No other characters are valid. Double quotation marks are removed before processing the database name. They cannot be used to embed other characters in the name.

Lowercase characters are not treated with special significance. They are considered the same as their uppercase counterparts.

For more information, see the *Oracle8i Administrator's Guide*.

DB_WRITER_PROCESSES

Parameter type:	integer
Parameter class:	static
Default value:	1
Range of values:	1-10

DB_WRITER_PROCESSES specifies the initial number of database writer processes for an instance.

If you use DBWR_IO_SLAVES, only one database writer process will be used, regardless of the setting for DB_WRITER_PROCESSES.

DBLINK_ENCRYPT_LOGIN

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE / FALSE

DBLINK_ENCRYPT_LOGIN specifies whether attempts to connect to other Oracle Servers through database links should use encrypted passwords.

When you attempt to connect to a database using a password, Oracle encrypts the password before sending it to the database. If the DBLINK_ENCRYPT_LOGIN parameter is TRUE, and the connection fails, Oracle does not re-attempt the connection. If this parameter is FALSE, Oracle re-attempts the connections using an unencrypted version of the password.

For more information, see the *Oracle8i Administrator's Guide*.

DBWR_IO_SLAVES

Parameter type:	integer
Parameter class:	static
Default value:	0
Range of values:	0 to system-dependent value

DBWR_IO_SLAVES specifies the number of I/O slaves used by the DBWn process. The DBWn process and its slaves always write to disk. By default, the value is 0 and I/O slaves are not used.

If DBWR_IO_SLAVES is set to a non-zero value, the numbers of I/O slaves used by the ARCH process LGWR process, and Recovery Manager are set to 4.

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

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

DELAYED_LOGGING_BLOCK_CLEANOUTS

Parameter type:	boolean
Parameter class:	static
Default value:	TRUE
Range of values:	TRUE / FALSE
OK to change:	yes
Multiple instances:	need not be identical

DELAYED_LOGGING_BLOCK_CLEANOUTS turns the delayed block cleanout feature on or off. This reduces pinging in an Oracle Parallel Server. Keeping this feature set to TRUE sets a fast path, no logging block cleanout at commit time. Logging the block cleanout occurs at the time of a subsequent change to the block. This generally improves Oracle Parallel Server performance, particularly if block pings are a problem.

When Oracle commits a transaction, each block that the transaction changed is not immediately marked with the commit time. This is done later, on demand, when the block is read or updated. This is called *block cleanout*.

When block cleanout is performed during an update to a current block, the cleanout changes and the redo records are appended with those of the update. In previous releases, when block cleanout was needed during a read to a current block, extra cleanout redo records were generated and the block was dirtied. This has been changed.

When a transaction commits, all blocks changed by the transaction are cleaned out immediately. This cleanout performed at commit time is a "fast version" which does not generate redo log records (*delayed logging*) and does not re-ping the block. Most blocks will be cleaned out in this way, with the exception of blocks changed by long running transactions.

During queries, therefore, the data block’s transaction information is normally up-to-date and the frequency of needing block cleanout is greatly reduced. Regular block cleanouts are still needed when querying a block where the transactions are still truly active, or when querying a block which was not cleaned out during commit.

Note: In long-running transactions, block cleanouts will not be performed during the transaction. If the transaction is not long running, block cleanout will be performed and the block cleanout is logged at the change of block.

During changes (INSERT, DELETE, UPDATE), the cleanout redo log records are generated and appended with the redo of the changes.

DISCRETE_TRANSACTIONS_ENABLED

Obsoleted in 8.1.3.

DISK_ASYNCH_IO

Parameter type:	boolean
Parameter class:	static
Default value:	TRUE
Range of values:	TRUE, FALSE

DISK_ASYNCH_IO can be used to control whether I/O to datafiles, controlfiles and logfiles are asynchronous. If a platform supports asynchronous I/O to disk, it is recommended that this parameter is left to its default. However, if the asynchronous I/O implementation is not stable, this parameter can be set to FALSE to disable asynchronous I/O. If a platform does not support asynchronous I/O to disk, this parameter will have no effect.

If DISK_ASYNCH_IO is set to FALSE, then DBWR_IO_SLAVES should also be set.

DISTRIBUTED_LOCK_TIMEOUT

Obsoleted in 8.1.3.

DISTRIBUTED_RECOVERY_CONNECTION_HOLD_TIME

Obsolete in 8.1.3.

DISTRIBUTED_TRANSACTIONS

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent
Range of values:	0 - TRANSACTIONS

DISTRIBUTED_TRANSACTIONS specifies the maximum number of distributed transactions in which this database can concurrently participate. The value of this parameter cannot exceed the value of the parameter TRANSACTIONS.

If network failures occur at an abnormally high rate, causing many in-doubt transactions, you may want to temporarily decrease this parameter's value. This limits the number of concurrent distributed transactions, which then reduces the number of in-doubt transactions. Thus, the amount of blocked data and possible heuristic decision making (because of in-doubt transactions) is reduced.

If DISTRIBUTED_TRANSACTIONS is set to 0, no distributed transactions are allowed for the database. The recovery (RECO) process also does not start when the instance starts up.

For more information, see the *Oracle8i Administrator's Guide* and *Oracle8i Distributed Database Systems*. See also your operating system-specific Oracle documentation for the default value.

DML_LOCKS

Parameter type:	integer
Parameter class:	static
Default value:	derived (4 * TRANSACTIONS)
Range of values:	20 - unlimited, 0
Multiple instances:	must all have positive values or must all be 0

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 3 users are modifying data in one table, then 3 entries would be required. If 3 users are modifying data in 2 tables, then 6 entries would be required.

The default value assumes an average of 4 tables referenced per transaction. For some systems, this value may not be enough.

If the value is set 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. If the value is set to 0 on one instance, it must be set to 0 on all instances of an Oracle Parallel Server.

For more information on data concurrency, see *Oracle8i Parallel Server Concepts and Administration*, *Oracle8i Concepts*, and *Oracle8i Distributed Database Systems.*

PDML Restrictions

DML_LOCKS has the following PDML restrictions regarding locks acquired by a parallel UPDATE / DELETE / INSERT statement.

- The coordinator acquires one Table lock SX where there is one Partition lock X per partition.
- For parallel UPDATE / DELETE, unless the UPDATE / DELETE's WHERE clause specifies the partitions involved, the coordinator will acquire partition locks for all partitions.
- For parallel INSERT, the coordinator will acquire partition locks for all partitions.
- Each slave acquires one Table lock SX where there is one Partition lock NULL per partition, and where there is one Partition-Wait lock X per partition.
- A slave can work on one or more partitions but a partition can only be worked on by one slave. So for a table with 600 partitions, running with parallel degree 100, assuming all partitions are involved in the parallel UPDATE / DELETE statement. There are the following requirements:
  - The coordinator acquires one Table lock SX and 600 Partition locks X
  - Total slaves acquire 100 Table locks SX, 600 Partition locks NULL, and 600 Partition-Wait locks X

ENQUEUE_RESOURCES

Parameter type:	integer
Parameter class:	static
Default value:	derived
Range of values:	10 - unlimited

An enqueue is a sophisticated locking mechanism which permits several concurrent processes to share known resources to varying degrees. Any object which 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 in latches there is no ordered queue of waiting processes as there are in enqueues. Processes waiting for latches can either use timers to wake up and retry or spin (only in multiprocessors).

ENQUEUE_RESOURCES sets the number of resources that can be concurrently locked by the lock manager. The default value of ENQUEUE_RESOURCES is derived from the SESSIONS parameter and should be adequate, as long as DML_LOCKS + 20 is less than ENQUEUE_RESOURCES. For three or fewer sessions, the default value is 20. For 4 to 10 sessions, the default value is ((SESSIONS - 3) * 5) + 20; and for more than 10 sessions, it is ((SESSIONS - 10) * 2) + 55.

If you explicitly set ENQUEUE_RESOURCES to a value higher than DML_LOCKS + 20, then the value you provide is used.

If there are many tables, the value may be increased. Allow one per resource (regardless of the number of sessions or cursors using that resource), not one per lock Only increase this parameter if Oracle returns an error specifying that enqueues are exhausted.

For more information on data concurrency, see *Oracle8i Parallel Server Concepts and Administration, Oracle8i Concepts* and *Oracle8i Distributed Database Systems.*

ENT_DOMAIN_NAME

Parameter type:	X.500 Distinguished Name
Parameter class:	static
Default value:	none
Range of values:	all X.500 Distinguished Name format values

Note: Global user functionality is currently available only to beta customers. This feature will be available to all users in a subsequent release of Oracle8i.

Use this parameter to indicate which Enterprise Domain (for Global User Security) the server belongs to. Enterprise Roles for the server will be searched for under this domain in the enterprise directory service. See the *Oracle Advanced Security Administrator's Guide* for more information.

EVENT

Parameter type:	string
Parameter class:	static
Default value:	NULL

EVENT is used to debug the system. This parameter should not usually be altered except at the direction of Oracle technical support personnel.

FAST_FULL_SCAN_ENABLED

Obsoleted in 8.1.3.

FAST_START_IO_TARGET

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM set at runtime
Default value:	all the buffers in the cache
Range of values:	1000 to all buffers in the cache, setting to 0 disables limiting recovery IOs
OK to change:	yes, at run-time
Multiple instances:	can have different values

FAST_START_IO_TARGET (*available only with the Oracle Enterprise Edition*) specifies the number of IOs that should be needed during crash or instance recovery. It imposes a more accurate bound on the number of recovery IOs than DB_BLOCK_MAX_DIRTY_TARGET.

When this parameter is set, DBWn writes dirty buffers out more aggressively to keep the number of blocks that must be processed during recovery below the value specified in the parameter. Note that this parameter does not impose a hard limit on the number of recovery IOs. There may be transient workload situations in which the number of IOs needed during recovery is greater than the value specified in this parameter, but if this occurs, DBWn *will not* slow down database activity.

Smaller values for this parameter result in faster recovery times. This improvement in recovery performance is achieved at the expense of additional writing activity during normal processing. See the *Oracle8i Backup and Recovery Guide* and V$INSTANCE_RECOVERY on page 3-51 for more information.

Setting this parameter’s value to 0 disables the mechanism that limits the number of IOs that need to be performed during recovery. All other writing activity is unaffected.

FAST_START_PARALLEL_ROLLBACK

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	low
Range of values:	false, low, high

FAST_START_PARALLEL_ROLLBACK helps to determine the maximum number of processes which may exist for performing parallel rollback. If the value is false, parallel rollback is disabled. If the value is low, 2 * CPU_COUNT number of processes may be used. If the value is high, at most 4 * CPU_COUNT number of rollback servers are used for parallel rollback.

FIXED_DATE

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	NULL

FIXED_DATE lets you set a constant date that SYSDATE will always return instead of the current date. The format of the date is:

```
YYYY-MM-DD-HH24:MI:SS.
```

It also accepts the default Oracle date format, without a time. Specify the value with double quotes (but not single quotes) or without quotes. For example,

```
FIXED_DATE = "30-nov-95"
```

or

```
FIXED_DATE = 30-nov-95
```

This parameter is useful primarily for testing.

FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	see below
Range of values:	TRUE, FALSE
OK to change:	yes
Multiple instances:	must have identical values

FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY is a Parallel Server parameter. The value of this parameter lets the database administrator control whether Oracle freezes the entire database during instance recovery. When this parameter is set to TRUE, Oracle freezes the entire database during instance recovery. The advantage of freezing the whole database is that it stops all other disk activities except those for instance recovery. This lets instance recovery complete faster. The drawback of freezing the whole database is that the entire database becomes unavailable during instance recovery.

When this parameter is set to FALSE, Oracle does not freeze the entire database, unless Oracle is responsible for *resilvering* some of the mirrored data files. Resilvering means ensuring data consistency of mirrored data files after a node crash. When Oracle does not freeze the entire database, part of the unaffected database will be accessible during instance recovery.

If all online datafiles use hash locks, the default value of this parameter is FALSE. If any data files use fine-grain locks, the default is TRUE.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

GC_DEFER_TIME

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	10
Range of values:	any positive integer
Multiple instances:	can have different values
OK to change:	no

GC_DEFER_TIME specifies the time (in 100ths of a second) that the server waits, or *defers*, before responding to forced-write requests for hot blocks from other instances. Specifying the GC_DEFER_TIME parameter makes it more likely that buffers will be properly cleaned out before being written, thus making them more useful when they are read by other instances. It also improves the chance of hot blocks being used multiple times within an instance between forced writes.

The default value, 0, means that the feature is disabled: no deferring occurs.

GC_FILES_TO_LOCKS

Parameter type:	string
Parameter class:	static
Default value:	NULL
Multiple instances:	must have identical values
OK to change:	yes

GC_FILES_TO_LOCKS is a Parallel Server parameter. This parameter controls the mapping of PCM locks to datafiles. The value of the parameter should be set to cover as many files as possible. Thus, to avoid performance problems, you should always change GC_FILES_TO_LOCKS when the size of datafiles change or when new datafiles are added. This requires you to shutdown and restart your parallel server.

GC_FILES_TO_LOCKS has the following syntax:

```
GC_FILES_TO_LOCKS = "{file_list=lock_count[!blocks][EACH]}[:]..."
```

where *file_list* is one or more datafiles listed by their file numbers, or ranges of file numbers, with comma separators:

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

and *lock_count* is the number of PCM locks assigned to *file_list*. If *lock_count* is set to 0, then fine-grain locking is used for these files.

A colon (:) separates each clause that assigns a number of PCM locks to *file_list*. The optional parameter *blocks*, specified with the "!" separator, indicates the number of contiguous blocks covered by one lock. The default is non-contiguous blocks. EACH specifies that each datafile in *file_list* is assigned a separate set of *lock_count* PCM locks. Spaces are not allowed within the quotation marks.

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 since there will be locks which are not covering any blocks.

The datafiles not specified in GC_FILES_TO_LOCKS are covered, by default, by releasable locks. Releasable locks are controlled by a different parameter, GC_RELEASABLE_LOCKS. See "GC_RELEASABLE_LOCKS" on page 1-46.

To find the correspondence between filenames and file numbers, query the data dictionary view DBA_DATA_FILES. See "DBA_DATA_FILES" on page 2-85.

GC_FILES_TO_LOCKS has no effect on an instance running in exclusive mode.

For more information on GC_FILES_TO_LOCKS, see *Oracle8i Parallel Server Concepts and Administration*.

GC_LATCHES

Obsoleted in 8.1.3.

GC_LCK_PROCS

Obsoleted in 8.1.3.

GC_RELEASABLE_LOCKS

Parameter type:	integer
Parameter class:	static
Default value:	Defaults to the number of buffers (DB_BLOCK_BUFFERS)
Range of values:	0 - DB_BLOCK_BUFFERS or higher
Multiple instances:	can have different values
OK to change:	yes

Lock elements can be fixed or non-fixed. Fixed lock elements are used by hashed PCM locks, in which the lock element name is preassigned. Non-fixed lock elements are used with fine-grain locking. If the GC_RELEASABLE_LOCKS parameter is set, its value is used to allocate space for fine-grain locking. There is no maximum value, except as imposed by memory restrictions.

This parameter is specific to the Oracle Parallel Server in shared mode.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

GC_ROLLBACK_LOCKS

Parameter type:	string
Parameter class:	static
Default value:	20
Multiple instances:	must have identical values
OK to change:	yes

GC_ROLLBACK_LOCKS is a Parallel Server parameter. This parameter specifies, for each rollback segment, the number of distributed locks available for simultaneously modified rollback segment blocks. The default is adequate for most applications.

These instance locks are acquired in exclusive mode by the instance that acquires the rollback segment. They are used to force the instance to write rollback segment blocks to disk when another instance needs a read-consistent version of a block.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

GLOBAL_NAMES

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	TRUE
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. Oracle recommends setting this parameter to TRUE to ensure the use of consistent naming conventions for databases and links.

If you use distributed processing, set GLOBAL_NAMES to TRUE to ensure a unique identifying name for your database in a networked environment.

For more information, see the *Oracle8i Administrator's Guide*.

HASH_AREA_SIZE

Parameter type:	integer
Parameter class:	dynamic, scope= ALTER SESSION
Default value:	2 * SORT_AREA_SIZE
Range of values:	0 - system-dependent value

HASH_AREA_SIZE specifies the maximum amount of memory, in bytes, to be used for hash joins. If this parameter is not set, its value defaults to twice the value of the SORT_AREA_SIZE parameter.

HASH_JOIN_ENABLED

Parameter type:	boolean
Parameter class:	dynamic, scope= ALTER SESSION
Default value:	TRUE
Range of values:	TRUE, FALSE

HASH_JOIN_ENABLED specifies whether the optimizer should consider using a hash join as a join method. When set to FALSE, hash join is turned off; that is, it is not available as a join method that the optimizer can consider choosing. When set to TRUE, the optimizer will compare the cost of a hash join to other types of joins, and choose it if it gives the best cost.

HASH_MULTIBLOCK_IO_COUNT

Parameter type:	integer
Parameter class:	dynamic, scope= ALTER SESSION, ALTER SYSTEM
Default value:	query dependent, appears as 0
Range of values:	operating system dependent

HASH_MULTIBLOCK_IO_COUNT specifies how many sequential blocks a hash join reads and writes in one IO. When operating in multi-threaded server mode, however, this parameter is ignored (a value of 1 is used even if you set the parameter to another value). Because Oracle computes the value for this parameter based on the query, you need not set the value for this parameter.

The maximum value for HASH_MULTIBLOCK_IO_COUNT varies by operating system. It is always less than the operating system's maximum I/O size expressed as Oracle blocks (max_IO_size/DB_BLOCK_SIZE).

This parameter strongly affects performance because it controls the number of partitions into which the input is divided. If you change the parameter value, try to make sure that the following formula remains true:

R / M <= Po2(M/C)

where:

R = size of(*left input to the join*)
M = HASH_AREA_SIZE * 0.9
Po2(*n*) = largest power of 2 that is smaller than *n*
C = HASH_MULTIBLOCK_IO_COUNT * DB_BLOCK_SIZE

HI_SHARED_MEMORY_ADDRESS

Parameter type:	integer
Parameter class:	static
Default value:	0

SHARED_MEMORY_ADDRESS and HI_SHARED_MEMORY_ADDRESS specify the SGA's starting address at runtime. These parameters are ignored on platforms which specify the SGA's starting address at linktime.

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.

HS_AUTOREGISTER

Parameter type:	boolean
Parameter class:	dynamic, scope= ALTER SYSTEM
Default value:	TRUE
Range of values:	TRUE, FALSE

HS_AUTOREGISTER enables or disables automatic self-registration of HS agents. When enabled, this entails uploading information into the server's data dictionary to describe a previously unknown agent class or a new agent version.

Recommended usage is to set this parameter to TRUE. The Oracle server incurs less overhead when establishing subsequent connections through the same agent if this self-registered information is available in the server's data dictionary.

If HS_AUTOREGISTER = FALSE is used as the normal operating mode, new agents or agent version will repeatedly upload on each new connection. Operations on the connection will work, but the server data dictionary will not be populated automatically. Database administrators should identify such situations by watching for upload entries in the server's alert log,

The corrective action is:

- ALTER SYSTEM SET HS_AUTOREGISTER = TRUE

- Establish foreign data source connection(s) to trigger uploads and update data dictionary
- If desired, ALTER SYSTEM SET HS_AUTOREGISTER = FALSE

IFILE

Parameter type:	string
Parameter class:	static
Default value:	NULL
Range of values:	valid parameter filenames
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
```

INSTANCE_GROUPS

Parameter type:	string LIST
Parameter class:	static
Allowable values:	a string of group names, separated by commas

INSTANCE_GROUPS is a Parallel Server parameter. It can be specified in parallel mode only. This parameter assigns the current instance to the specified groups. The value of INSTANCE_GROUPS must be a comma-separated list of instance groups. Instance groups are used when allocating query slaves for a parallel operation.

See also "PARALLEL_INSTANCE_GROUP" on page 1-96.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

INSTANCE_NAME

Parameter type:	string
Parameter class:	static
Default value:	the SID
Allowable values:	any alphanumeric characters

INSTANCE_NAME is a string value representing the name of the instance and is used to uniquely identify a specific instance when multiple instances share common services names. INSTANCE_NAME should not be confused with the SID, which actually uniquely identifies the instances shared memory on a host.

For more information, see *Oracle8i Parallel Server Concepts and Administration* and *Net8 Administrator's Guide*.

INSTANCE_NUMBER

Parameter type:	integer
Parameter class:	static
Default value:	lowest available number (depends on instance start up order and on the INSTANCE_NUMBER values assigned to other instances)
Range of values:	1 - maximum number of instances specified in CREATE DATABASE statement
Multiple instances:	if specified, instances must have different values
OK to change:	yes (can be specified in both parallel and exclusive modes)

INSTANCE_NUMBER is a Parallel Server parameter. This parameter can be specified in parallel mode or exclusive mode. It specifies a unique number that maps the instance to one group of free space lists for each table created with storage option FREELIST GROUPS.

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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

JAVA_POOL_SIZE

Parameter type:	string
Parameter class:	static
Default value:	10 MB

JAVA_POOL_SIZE specifies the size in bytes of the Java pool.

JOB_QUEUE_INTERVAL

Parameter type:	integer
Parameter class:	static
Default value:	60 (seconds)
Range of values:	1 - 3600 (seconds)
Multiple instances:	can have different values

JOB_QUEUE_INTERVAL specifies the interval between wake-ups for the SNP*n* background processes of the instance.

JOB_QUEUE_PROCESSES

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	0
Range of values:	0 - 36
Multiple instances:	can have different values

JOB_QUEUE_PROCESSES specifies the number of SNP*n* background processes per instance, where *n* is 0 to 9 followed by A to Z. Job queue processes are used to process requests created by DBMS_JOB.

Some job queue requests are created automatically; an example is refresh support for table snapshots. If you wish to have your table snapshots updated automatically, you must set JOB_QUEUE_PROCESSES to a value of one or higher.

For more information on managing table snapshots, see *Oracle8i Replication*.

LARGE_POOL_MIN_ALLOC

Obsoleted in 8.1.3.

LARGE_POOL_SIZE

Parameter type:	string
Parameter class:	static
Default value:	0, unless parallel execution or DBWR_IO_SLAVES are configured
Range of values:	minimum: 600K maximum: at least 2GB (maximum is operating system-specific)

LARGE_POOL_SIZE lets you specify the size of the large pool allocation heap.

The Large Pool allocation heap is used by the Multi-Threaded Server for session memory, by Parallel Execution for message buffers, and by Backup for disk I/O buffers.

Parallel Execution allocates buffers out of the Large Pool *only* when PARALLEL_AUTOMATIC_TUNING is set to TRUE.

If LARGE_POOL_SIZE is left unset and the pool is required by parallel execution, then Oracle will compute a value automatically. The computation will be based on PARALLEL_MAX_SERVERS, PARALLEL_THREADS_PER_CPU, PARALLEL_SERVER_INSTANCES, MTS_DISPATCHERS, and DBWR_IO_SLAVES.

The computation will add 250k per session for the MTS server if MTS_DISPATCHERS is configured. A final computation will add a port-specific amount of memory for Backup I/O buffers.

Warning: this default computation can yield a size that is either too large to allocate or causes performance problems. In that case, you should set LARGE_POOL_SIZE to a number sufficiently small so that the database can start.

The value of the parameter can be specified in megabytes or kilobytes. LARGE_POOL_SIZE can accept a numerical value or a number followed by the suffix "K" or "M" where "K" means "multiply by 1000" and "M" means "multiply by 1000000".

See Oracle8i Tuning and Oracle8i Migration for further information.

LGWR_IO_SLAVES

Obsoleted in 8.1.3.

LICENSE_MAX_SESSIONS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	0
Range of values:	0 - number of session licenses
Multiple instances:	can have different values

LICENSE_MAX_SESSIONS specifies the maximum number of concurrent user sessions allowed simultaneously. When this limit is reached, only users with the RESTRICTED SESSION privilege can connect to the server. 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 non-zero number, you might also want to set LICENSE_SESSIONS_WARNING.

Concurrent usage licensing and user licensing should not both be enabled. Either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS should always be zero.

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.

For more information, see the *Oracle8i Administrator's Guide*.

LICENSE_MAX_USERS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	0
Range of values:	0 - number of user licenses
Multiple instances:	should have the same values

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.

Concurrent usage (session) licensing and user licensing should not both be enabled. Either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS, or both, should be zero.

If different instances specify different values for this parameter, the value of the first instance to mount the database takes precedence.

For more information, see the *Oracle8i Administrator's Guide*.

LICENSE_SESSIONS_WARNING

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	0

Range of values:	0 - LICENSE_MAX_SESSIONS
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 when approaching the concurrent usage (session) limit. If you set this parameter to a nonzero number, you should also set LICENSE_MAX_SESSIONS.

For more information, see the *Oracle8i Administrator's Guide*.

LM_LOCKS

Parameter type:	integer
Parameter class:	static
Default value:	12000
Range of values:	minimum: 512, maximum: limited by ■ the shared memory available in the operating system ■ the maximum size of contiguous shared memory segment; otherwise, it is limited only by the address space
Multiple instances:	must have the same value

LM_LOCKS is a Parallel Server parameter. This parameter specifies the number of locks which will be configured for the lock manager. The number of locks can be represented by the following equation, where R is the number of resources, N is the total number of nodes, and L is the total number of locks:

$$L = R + (R*(N - 1)) / N$$

Note that lock configurations are per lock manager instance. Thus the value of LM_LOCKS must be the same for all lock manager instances. Also note that, in the worst case, up to 2 * GC_RELEASABLE_LOCKS could be required with DBA locking.

LM_PROCS

Parameter type:	integer
Parameter class:	static
Default value:	64 + the maximum number of instances supported on the port
Range of values:	minimum: 36 maximum: the result of the following equation: PROCESSES + maximum number of instances + safety factor Note: This assumes that the PROCESSES parameter has already included the Oracle background processes, including LMON and LMD0. The safety factor should be added to account for temporary overhead or unavailability of some procedure during the clean-up of dead processes.
Multiple instances:	must have the same value

LM_PROCS is a Parallel Server parameter. The value of this parameter represents the value of the PROCESSES parameter plus the maximum number of instances. Note that the processes configurations are per lock manager instance. Thus, the value for LM_PROCS must be the same for all lock manager instances.

LM_RESS

Parameter type:	integer
Parameter class:	static
Default value:	6000
Range of values:	minimum: 256 maximum: limited by ▪ the shared memory available in the operating system ▪ the maximum size of contiguous shared memory segment; otherwise, it is limited only by the address space
Multiple instances:	must have the same value

LM_RESS is a Parallel Server parameter. This parameter controls the number of resources that can be locked by each lock manager instance. It is recommended that each instance be assigned the same parameter value.

The value specified for LM_RESS should be less than 2 * DML_LOCKS plus an overhead of about 20 locks. However, with DBA locking, up to 2 * GC_RELEASABLE_LOCKS resources to cover all the BL locks, at least in the worst case.

LM_RESS covers the number of lock resources allocated for DML, DDL (data dictionary locks), data dictionary and library cache locks plus the file and log management locks.

LOCAL_LISTENER

Parameter type:	string
Parameter class:	static
Default value:	"(ADDRESS_LIST = (Address = (Protocol = TCP) (Host=localhost) (Port=1521)) (Address=(Protocol = IPC) (Key= *DBname*)))"

The LOCAL_LISTENER parameter is optional and identifies "local" Net8 listeners. LOCAL_LISTENER specifies the network name of either a single address or an address list of Net8 listeners. These Net8 listeners need to be running on the same machine as the instance.

The instance and dispatchers register certain information with the listener. This information enables the listener to connect clients to the appropriate dispatchers and dedicated servers. In order to connect clients to dedicated servers, the listener and the instance must be running on the same machine.

When it is present, the LOCAL_LISTENER parameter overrides the obsolete MTS_LISTENER_ADDRESS and MTS_MULTIPLE_LISTENERS parameters if specified. For more information on these parameters, see "MTS_LISTENER_ADDRESS" on page 1-75 and "MTS_MULTIPLE_LISTENERS" on page 1-75.

For more information about instances, listener processes, and dispatcher processes, see the *Oracle8i Administrator's Guide*. See your operating system-specific Oracle documentation and Net8 documentation for a description of how to specify addresses for the protocols on your system.

LOCK_NAME_SPACE

Parameter type:	string
Parameter class:	static
Range of values:	eight characters maximum, no special characters allowed

LOCK_NAME_SPACE specifies the name space that the distributed lock manager (DLM) uses to generate lock names. This might need to be set if there is a standby or clone database with the same database name on the same cluster.

LOCK_SGA

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

LOCK_SGA is used to lock the entire SGA into physical memory. It is ignored on platforms that don't support it.

LOCK_SGA_AREAS

Obsoleted in 8.1.3.

LOG_ARCHIVE_BUFFER_SIZE

Obsoleted in 8.1.3.

LOG_ARCHIVE_BUFFERS

Obsoleted in 8.1.3.

LOG_ARCHIVE_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	NULL string default
Range of values:	either a NULL string or any valid path or device name, except raw partitions
Multiple instances:	can have different values

Deprecated in favor of LOG_ARCHIVE_DEST_n when Oracle Enterprise Edition is installed. If Oracle Enterprise Edition is not installed or it is installed but you have not specified LOG_ARCHIVE_DEST_n, this parameter can be used as described below.

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 non_NULL string value. 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 command ARCHIVE LOG START *filespec* for automatic archiving, where *filespec* is the new archive destination. To permanently change the destination, use the command ALTER SYSTEM SET LOG_ARCHIVE_DEST = filespec, where filespec is the new archive destination.

For more information, see the *Oracle8i Backup and Recovery Guide*. See also "LOG_ARCHIVE_DUPLEX_DEST" on page 1-63 on page 1-63, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-65, and "V$ARCHIVE_DEST" on page 3-5.

See 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
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	NULL
Range of values:	valid keyword definitions

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

LOG_ARCHIVE_DEST_n defines a destination and attributes for the archive redo log file group. The parameter suffix (1 through 5) specifies one of the 5 corresponding LOG_ARCHIVE_DEST_n destination parameters. The parameter number suffix is defined as the ’handle’ displayed by the fixed-table queries.

```
LOG_ARCHIVE_DEST_n = ("null_string" | SERVICE=tnsnames-service |
                     LOCATION=directory-spec)[MANDATORY | OPTIONAL]
                     [REOPEN=integer]
```

SERVICE specifies a standby destination. Net8 (IPC or TCP) will be used to transmit the archivelog. There must be a standby instance associated with the destination. `tnsnames-service` corresponds to an appropriate service-name in TNSNAMES.ORA.

LOCATION specifies a local file-system destination.

MANDATORY specifies that archiving to the destination must succeed before the REDO logfile can be made available for re-use.

OPTIONAL specifies that successful archiving to the destination is not required before the REDO log file can be made available for re-use. If the "must succeed count" (LOG_ARCHIVE_MIN_SUCCEED_DEST) is met the REDO logfile is marked for re-use. This is the default

REOPEN specifies an interval of time (in seconds) that must pass after an error has been encountered during archiving to the destination before future archives to the destination can be attempted. Future attempts are made when the next REDO logfile is archived. If a destination is MANDATORY Oracle recommend that you specify a REOPEN time that reduces the possibility of primary database shutdown due to lack of available online REDO logfiles.

Following is an example scenario:

```
LOG_ARCHIVE_MIN_SUCCEED_DEST=2

LOG_ARCHIVE_DEST_1='SERVICE=standby1 OPTIONAL REOPEN=120'
LOG_ARCHIVE_DEST_2='LOCATION=filespec MANDATORY REOPEN=5'
LOG_ARCHIVE_DEST_3='SERVICE=standby2 OPTIONAL REOPEN=60'

LOG_ARCHIVE_DEST_STATE_1=enable
LOG_ARCHIVE_DEST_STATE_2=enable
LOG_ARCHIVE_DEST_STATE_3=enable
```

In the above example, destination 1 and 3 are standby destinations, both are optional. Destination 1 has a reopen interval of 2 minutes, destination 3 has a reopen interval of 1 minute. Destination 2 is a local destination, completion is mandatory and a five second reopen interval is specified.

All three destinations are enabled and therefore available to ARCHIVELOG operations as target destinations.

The minimum number of destinations that must archive successfully for the redo log to be marked for re-use is set to two. This means, in addition to destination 2 (the mandatory destination) either destination 1 and/or destination 3 must also archive successfully.

LOG_ARCHIVE_DEST_STATE_n

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	ENABLE
Range of values:	ENABLE, DEFER

LOG_ARCHIVE_DEST_STATE_n specifies the availability state of the corresponding destination. The parameter suffix (1 through 5) specifies one of the 5 corresponding LOG_ARCHIVE_DEST_n destination parameters.

If enabled, a valid log archive destination can be used for a subsequent archiving operation (automatic or manual). If deferred, any valid destination information and attributes are preserved, but the destination is excluded from archiving operations until re-enabled.

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.

Changed by ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_n = value or ALTER SESSION SET LOG_ARCHIVE_DEST_STATE_n = value. For example:

```
ALTER SESSION SET LOG_ARCHIVE_DEST_STATE_n = enable
```

ALTER SESSION effectively hides the system level value for the session issuing the command. The system level value can only be reestablished by an ALTER SESSION using the current system level value.

Always refer to V$ARCHIVE_DEST for values in use for the current session. Corresponds to V$ARCHIVE_DEST index n.

LOG_ARCHIVE_DUPLEX_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	a NULL string
Range of values:	Either a NULL string or any valid path or device name, except raw partitions

Deprecated in favor of LOG_ARCHIVE_DEST_n when Oracle Enterprise Edition is installed. If Oracle Enterprise Edition is not installed, 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.

If LOG_ARCHIVE_DUPLEX_DEST is set to be a NULL string ("") or (’’), it means there is no duplex archive destination. The default of this parameter is a NULL string.

For more information, see "LOG_ARCHIVE_DEST_n" on page 1-61, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-65, and "V$ARCHIVE_DEST" on page 3-5.

LOG_ARCHIVE_FORMAT

Parameter type:	string
Parameter class:	static
Default value:	operating system-dependent (length for uppercase variables is also operating system-dependent)
Range of values:	any valid filename
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
%t     thread number
```

Using uppercase letters (for example, %S) for the variables causes the value to be a fixed length padded to the left with zeros.

The following is an example of specifying the archive redo log filename format:

```
LOG_ARCHIVE_FORMAT = "LOG%s_%t.ARC"
```

For more information, see *Oracle8i Administrator's Guide*. See also 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
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	1
Range of values:	any integer from 1 - 10 inclusive

LOG_ARCHIVE_MAX_PROCESSES specifies the number of ARCH processes to be invoked. This value is evaluated at instance startup if the LOG_ARCHIVE_START initialization parameter has the value TRUE; otherwise, this parameter is evaluated when the ARCH process is invoked via SQL*Plus or SQL syntax.

LOG_ARCHIVE_MIN_SUCCEED_DEST

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	1
Range of values:	1 - 5 (restricted to 1-2 when used with 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. When automatic archiving is enabled, the value generally cannot be more than the total number of destinations, and the number of enabled, valid MANDATORY destinations plus the number of enabled, valid non-standby OPTIONAL destinations.

If the value is less than the number of enabled, valid MANDATORY destinations, it will be 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 will essentially be treated as MANDATORY.

ALTER SESSION SET LOG_ARCHIVE_MIN_SUCCEED_DEST = n cannot be used when LOG_ARCHIVE_DEST or LOG_ARCHIVE_DUPLEX_DEST are in use.

Changed by ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST = number or ALTER SESSION SET LOG_ARCHIVE_MIN_SUCCEED_DEST = number. ALTER SESSION effectively hides the system level value for the session issuing the command. The system level value can only be re-established by an ALTER SESSION using the current system level value.

LOG_ARCHIVE_MIN_SUCCEED_DEST can be set to 1, and all destinations can be set to the null string as a necessary transitory state used to dynamically switch between the destinations specified by the LOG_ARCHIVE_DEST, LOG_ARCHIVE_DUPLEX_DEST parameters and the destinations specified by the LOG_ARCHIVE_DEST_n parameters.

For more information, see "LOG_ARCHIVE_DEST_n" on page 1-61, "LOG_ARCHIVE_DUPLEX_DEST" on page 1-63, and "V$ARCHIVE_DEST" on page 3-5.

LOG_ARCHIVE_START

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE / FALSE
Multiple instances:	can have different values

LOG_ARCHIVE_START is applicable only when you use the redo log in ARCHIVELOG mode, LOG_ARCHIVE_START indicates whether archiving should be automatic or manual when the instance starts up. TRUE indicates that archiving is automatic. FALSE indicates that the database administrator will archive filled redo log files manually. (The SQL*Plus command ARCHIVE LOG START or STOP overrides this parameter.)

In ARCHIVELOG mode, if all online redo log files fill without being archived, an error message is issued, and instance operations are suspended 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.

For more information, see the *Oracle8i Administrator's Guide*.

LOG_BLOCK_CHECKSUM

Obsoleted in 8.1.3.

LOG_BUFFER

Parameter type:	integer
Parameter class:	static
Default value:	maximum: 500K or 128K * CPU_COUNT, whichever is greater
Range of values:	operating system-dependent

LOG_BUFFER specifies the amount of memory, in bytes, that is used 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, the value 65536 or higher would not be unreasonable.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value and range of values.

LOG_CHECKPOINT_INTERVAL

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	operating system-dependent
Range of values:	unlimited (operating-system blocks, not database blocks)
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 are written between consecutive checkpoints.

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

Extremely frequent checkpointing can cause excessive writes to disk, possibly impacting transaction performance. In addition, if the intervals are so close together that the interval checkpoint requests are arriving at a rate faster than the rate at which Oracle can satisfy these requests, Oracle can choose to ignore some of these requests in order to avoid excessive interval checkpointing activity.

The number of times DBWn has been notified to do a checkpoint for a given instance is shown in the cache statistic **DBWR checkpoints**, which is displayed in the System Statistics Monitor of the Enterprise Manager. For more information about this statistic, see "DBWR checkpoints" in Appendix C, "Statistics Descriptions".

Note that specifying a value of 0 (zero) for LOG_CHECKPOINT_INTERVAL causes the parameter to be ignored. Only non-zero values of this parameter are considered meaningful. A LOG_CHECKPOINT_INTERVAL value of 0 has the same effect as setting the parameter to infinity.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value.

LOG_CHECKPOINT_TIMEOUT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	Oracle8*i*: 900 seconds. Enterprise Edition: 1800 seconds.
Range of values:	0 - unlimited
Multiple instances:	can have different values

LOG_CHECKPOINT_TIMEOUT specifies the maximum amount of time before another checkpoint occurs. The value is specified in seconds.

The time begins at the start of the previous checkpoint, then a checkpoint occurs after the amount of time specified by this parameter. Specifying a value of 0 for the timeout disables time-based checkpoints. Hence, setting the value to 0 is not recommended.

Note: 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.

For more information, see the *Oracle8i Administrator's Guide.*

LOG_CHECKPOINTS_TO_ALERT

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE / FALSE

LOG_CHECKPOINTS_TO_ALERT allows you to log your checkpoints to the alert file. This parameter is useful to determine if checkpoints are occurring at the desired frequency.

For more information, see *Oracle8i Backup and Recovery Guide*.

LOG_FILE_NAME_CONVERT

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	none
Range of values:	character strings

The value of 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. Adding a log file to the primary database necessitates adding a corresponding file to the standby database.

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

Set the value of this parameter to two strings: 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.

LOG_FILE_NAME_CONVERT should also be used to rename the logfiles in the clone controlfile when setting up the clone database during Tablespace Point-In-Time Recovery. For more information, see Oracle8i Backup and Recovery Guide.

LOG_FILES

Obsoleted in 8.1.3.

LOG_SIMULTANEOUS_COPIES

Obsoleted in 8.1.3.

LOG_SMALL_ENTRY_MAX_SIZE

Obsoleted in 8.1.3.

MAX_COMMIT_PROPAGATION_DELAY

Parameter type:	integer
Parameter class:	static
Default value:	700
Range of values:	0 - 90000
Multiple instances:	must have identical values
OK to change:	no

MAX_COMMIT_PROPAGATION_DELAY is a Parallel Server parameter. This initialization parameter should not be changed except under a limited set of circumstances specific to the Parallel Server.

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 LGWR. It determines if 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.

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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

MAX_DUMP_FILE_SIZE

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	10000 blocks
Range of values:	0 - UNLIMITED

MAX_DUMP_FILE_SIZE specifies the maximum size of trace files to be written. Change this limit if you are concerned that trace files may take up too much space.

MAX_DUMP_FILE_SIZE can accept a numerical value or a number followed by the suffix "K", or "M", where "K" means multiply by 1000 and "M" means multiply by 1000000. A numerical value for MAX_DUMP_FILE_SIZE specifies the maximum size in operating system blocks, whereas a number followed by a "K" or "M" suffix specifies the file size in number of bytes. MAX_DUMP_FILE_SIZE can also assume the special value string UNLIMITED. UNLIMITED means that there is no upper limit on trace file size, thus dump files can be as large as the operating system permits.

For more information, see the *Oracle8i Administrator's Guide*.

MAX_ENABLED_ROLES

Parameter type:	integer
Parameter class:	static
Default value:	20
Range of values:	0 - 148

MAX_ENABLED_ROLES specifies the maximum number of database roles that a user can enable, including sub-roles.

The actual number of roles a user 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 SCOTT can have 7 roles enabled, the five enabled by MAX_ENABLED_ROLES plus PUBLIC and SCOTT.

For more information, see the *Oracle8i Administrator's Guide*.

MAX_ROLLBACK_SEGMENTS

Parameter type:	integer
Parameter class:	static
Default value:	max(30, TRANSACTIONS / TRANSACTIONS_PER_ ROLLBACK_SEGMENT)
Range of values:	2 - 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 INUSE) simultaneously by one instance. For more information, see the *Oracle8i Administrator's Guide*.

MAX_TRANSACTION_BRANCHES

Obsoleted in 8.1.3.

MTS_DISPATCHERS

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	NULL

The parsing software supports a name-value syntax to enable the specification of attributes in a position-independent case-insensitive manner. For example:

```
MTS_DISPATCHERS = "(PROTOCOL=TCP)(DISPATCHERS=3)"
```

One and only one of the following attributes is required: ADDRESS, DESCRIPTION, or PROTOCOL.

Attribute	Description
ADDRESS (ADD or ADDR)	The network address (in Net8 syntax) of the end point which the dispatchers will listen on. (Includes the protocol.)
DESCRIPTION (DES or DESC)	The network description (in Net8 syntax) of the end point which the dispatchers will listen on. (Includes the protocol.)

PROTOCOL (PRO or PROT)	The network protocol for which the dispatchers will generate a listening end point.

The ADDRESS and DESCRIPTION attributes provides support for the specification of additional network attributes. (This enables support of multi-homed hosts.)

The attributes CONNECTIONS, DISPATCHERS, LISTENER, MULTIPLEX, POOL, SERVICE, and TICKS are optional:

Attribute	Description
CONNECTIONS (CON or CONN)	The maximum number of network connections to allow for each dispatcher. Default is set by Net8 and is platform specific.
DISPATCHERS (DIS or DISP)	The initial number of dispatchers to start. Default is 1.
LISTENER (LIS, LIST)	The network name of an address or address list of the Net8 listeners with which the dispatchers will register. The LISTENER attribute makes it easier to administer multi-homed hosts. This attribute specifies the appropriate listeners with which the dispatchers will register. The LISTENER attribute overrides the LOCAL_LISTENER parameter.
MULTIPLEX (MUL or MULT)	Used to enable the Net8 "Network Session Multiplex" feature. If "1", "ON", "YES", "TRUE", or "BOTH" is specified, then "Network Session Multiplex" is enabled for both incoming and outgoing network connections. If "IN" is specified, then "Network Session Multiplex" is enabled for incoming network connections. If "OUT" is specified, then "Network Session Multiplexing" is enabled for outgoing network connections. If "0", "NO", "OFF", or "FALSE" is specified, then "Network Session Multiplexing" is disabled for both incoming and outgoing network connections. The default "Network Session Multiplex" is disabled on both incoming and outgoing network connections.
PRESENTATION (PRE or PRES)	Used to enable support of specific presentation protocols. If GIOP is specified, the dispatcher will listen on the specified protocol for GIOP messages. For IIOP support, the protocol specified must be TCP. The default presentation is "TTC".

Attribute	Description
POOL (POO)	Used to enable the Net8 "Connection Pooling" feature. If a number is specified, then "Connection Pooling" is enabled for both incoming and outgoing network connections and the number specified is the timeout in ticks for both incoming and outgoing network connections. If "ON", "YES", "TRUE", or "BOTH" is specified, then "Connection Pooling" is enabled for both incoming and outgoing network connections and the default timeout (set by Net8) will be used for both incoming and outgoing network connections. If "IN" is specified, then "Connection Pooling" is enabled for incoming network connections and the default timeout (set by Net8) will be used for incoming network connections. If "OUT" is specified, then "Connection Pooling" is enabled for outgoing network connections and the default timeout (set by Net8) will be used for outgoing network connections. If "NO", "OFF", or "FALSE" is specified, then "Connection Pooling" is disabled for both incoming and outgoing network connections. POOL can also be assigned a name-value string such as: "(IN=10)", "(OUT=20)", or "(IN=10)(OUT=20)", in which case, 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 (set by Net8) will be used. The default "Connection Pooling" is disabled on both incoming and outgoing network connections.
SERVICE (SER, SERV)	The service name which the dispatchers register with the Net8 listeners. The SERVICE attribute overrides the MTS_SERVICE parameter. This attribute specifies a service name that the dispatchers will use to register. For more information, see "MTS_SERVICE".
SESSIONS (SES or SESS)	The maximum number of network sessions to allow for each dispatcher. Default is set by Net8 and is platform-specific.
TICKS (TIC or TICK)	The size of a network tick in seconds. See the *Net8 Administrator's Guide* for more details about what this means. The default is set by Net8 and is platform-specific.

For more information, see the *Oracle8i Administrator's Guide*. See also the *Net8 Administrator's Guide*.

MTS_LISTENER_ADDRESS

Obsoleted in 8.1.3.

MTS_MAX_DISPATCHERS

Parameter type:	integer
Parameter class:	static
Default value:	if dispatchers are configured, then defaults to whichever is greater: 5 or the number of dispatchers configured
Range of values:	operating system-dependent

MTS_MAX_DISPATCHERS specifies the maximum number of dispatcher processes allowed to be running simultaneously.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value and range of values.

MTS_MAX_SERVERS

Parameter type:	integer
Parameter class:	static
Default value:	20
Range of values:	operating system-dependent

MTS_MAX_SERVERS specifies the maximum number of shared server processes allowed to be running simultaneously.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value and range of values.

MTS_MULTIPLE_LISTENERS

Obsoleted in 8.1.3.

MTS_RATE_LOG_SIZE

Obsoleted in 8.1.3.

MTS_RATE_SCALE

Obsoleted in 8.1.3.

MTS_SERVERS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM,
Default value:	1
Range of values:	operating system-dependent

MTS_SERVERS specifies the number of server processes that you want to create when an instance is started up.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the default value and range of values.

MTS_SERVICE

Obsoleted in 8.1.3.

NLS_CALENDAR

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	Gregorian
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, if NLS_CALENDAR is set to "Japanese Imperial", the date format is "E YY-MM-DD", and the date is May 15, 1997, then the SYSDATE is displayed as follows:

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

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_COMP

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	BINARY
Range of values:	any valid character string, with a maximum of 10 bytes (not including null)

NLS_COMP lets you avoid the cumbersome process of using NLS_SORT in SQL statements. Normally, comparison in the WHERE clause is binary. To use linguistic comparison, the NLSSORT function must be used. Sometimes this can be tedious, especially when the linguistic sort needed has already been specified in the NLS_SORT session parameter. NLS_COMP can be used in such cases to indicate that the comparisons must be linguistic according to the NLS_SORT session parameter.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_CURRENCY

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	derived
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.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_DATE_FORMAT

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	derived
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"
```

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_DATE_LANGUAGE

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	value for NLS_LANGUAGE
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 (AM, PM, AD, BC). The default value of this parameter is the language specified by NLS_LANGUAGE.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_DUAL_CURRENCY

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	Dual currency symbol
Range of values:	any valid format name

NLS_DUAL_CURRENCY can be used to override the default dual currency symbol defined in the territory. When starting a new session without setting NLS_DUAL_CURRENCY, the default dual currency symbol defined in the territory of your current language environment will be used. When you set NLS_DUAL_CURRENCY, you will start up a session with its value as the dual currency symbol.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_ISO_CURRENCY

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	derived
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. The default value of this parameter is determined by NLS_TERRITORY.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_LANGUAGE

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	operating system-dependent
Range of values:	any valid language name

NLS_LANGUAGE specifies the default language of the database. This language is used for messages, the day and month names, the symbols for AD, BC, AM, and PM, and the default sorting mechanism. This parameter has the format:

```
NLS_LANGUAGE = FRENCH
```

Examples of supported languages are American, French, and Japanese.

This parameter determines the default values of the parameters NLS_DATE_LANGUAGE and NLS_SORT. For a complete list of languages, see Oracle8i National Language Support Guide.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*, your country release notes, and operating system-specific Oracle documentation.

NLS_NUMERIC_CHARACTERS

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	derived

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

Any character can be the decimal or group separator. The two characters specified must be single-byte, and both characters must be different from each other each other. The characters cannot be any numeric character or any of the following characters: plus (+), hyphen (-), less than sign (<), greater than sign (>).

The characters are specified in the following format:

```
NLS_NUMERIC_CHARACTERS = "<decimal_character><group_separator>"
```

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 = ", "
```

The default value of this parameter is determined by NLS_TERRITORY.

For more information, see *Oracle8i National Language Support Guide*. See also the *Oracle8i Administrator's Guide*.

NLS_SORT

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	derived
Range of values:	BINARY or 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 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 into the execution plan.

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

The default value of this parameter depends on the value of the NLS_LANGUAGE parameter.

For more information on this parameter, see *Oracle8i National Language Support Guide* and the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the sorting rules used by the linguistic sorting mechanisms.

NLS_TERRITORY

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	operating system-dependent
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. Also specifies the default date format, the default decimal character and group separator, and the default ISO and local currency symbols. Supported territories include America, France, Japan, and so on. For a complete list of territories, see Oracle8i National Language Support Guide.

This parameter determines the default values for the following parameters: NLS_CURRENCY, NLS_ISO_CURRENCY, NLS_DATE_FORMAT, and NLS_NUMERIC_CHARACTERS.

For more information, see *Oracle8i National Language Support Guide* and the *Oracle8i Administrator's Guide*. See your operating system-specific Oracle documentation for the territory-dependent default values for these parameters.

OBJECT_CACHE_MAX_SIZE_PERCENT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	10%
Range of values:	0% to operating system-dependent maximum

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.

OBJECT_CACHE_OPTIMAL_SIZE

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	100 Kbytes
Range of values:	10 Kbytes to operating system-dependent maximum

OBJECT_CACHE_OPTIMAL_SIZE specifies the size to which the session object cache is reduced when the size of the cache exceeds the maximum size.

OGMS_HOME

Obsoleted in 8.1.3.

OPEN_CURSORS

Parameter type:	integer
Parameter class:	static
Default value:	64
Range of values:	1 - operating system limit

OPEN_CURSORS specifies the maximum number of open cursors (context areas) a session can have at once. This constrains a session from opening an excessive number of cursors. Assuming that a session does not open the number of cursors specified by OPEN_CURSORS, there is no added overhead by setting this value too high.

It is important to have the value of OPEN_CURSORS set high enough to prevent your application from running out of open cursors. The number will vary from one application to another.

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

For more information, see the *Oracle8i Administrator's Guide.* See also your operating system-specific Oracle documentation for the range of values.

OPEN_LINKS

Parameter type:	integer
Parameter class:	static
Default value:	4
Range of values:	0 - 255

OPEN_LINKS specifies the maximum number of concurrent open connections to remote databases in one session. The value should equal or exceed the number of databases referred to in a single SQL statement that references multiple databases so that all the databases can be open to execute the statement. Value should be increased if many different databases are accessed over time. Thus, if queries alternately access databases A, B, and C and OPEN_LINKS is set to 2, time would be spent waiting while one connection was broken and another made.

This parameter refers only to connections used for distributed transactions. Direct connections to a remote database specified as an application connects are not counted. For information on migratable open connections for XA transactions, see "OPEN_LINKS_PER_INSTANCE".

If OPEN_LINKS is set to 0, then no distributed transactions are allowed.

For more information, see the *Oracle8i Administrator's Guide*

OPEN_LINKS_PER_INSTANCE

Parameter type:	integer
Parameter class:	static
Default value:	4
Range of values:	0 - UB4MAXVAL
Multiple instances	can be specified for multiple instances. The value need not be same for all instances
OK to change:	yes, provided a shutdown and restart is performed

OPEN_LINKS_PER_INSTANCE specifies the maximum number of migratable open connections. 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 that created the connection is the same as the user that owns the transaction.

OPEN_LINKS_PER_INSTANCE is different from the OPEN_LINKS parameter in that OPEN_LINKS indicates the number of connections from a session. The OPEN_LINKS parameter is not applicable to XA applications. For more information, see "OPEN_LINKS" on page 1-84.

OPS_ADMIN_GROUP

Obsoleted in 8.1.3.

OPTIMIZER_FEATURES_ENABLE

Parameter type:	string
Parameter class:	static
Default value:	8.0.0
Range of values:	8.0.0; 8.0.3; 8.0.4; 8.1.3

OPTIMIZER_FEATURES_ENABLE allows you to change the init.ora parameters which control the optimizer's behavior. The parameters affected are PUSH_JOIN_PREDICATE, FAST_FULL_SCAN_ENABLED, COMPLEX_VIEW_MERGING, and B_TREE_BITMAP_PLANS. The values 8.0.0 and 8.0.3 set those parameters to FALSE; 8.0.4 sets them to TRUE. However, regardless of the setting, you can change each parameter individually.

OPTIMIZER_INDEX_CACHING

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	0
Range of values:	0-100

OPTIMIZER_INDEX_CACHING lets the user adjust the behavior of the cost-based optimizer to select nested loops joins more often. The cost of executing a nested loops join where an index is used to access the inner table is highly dependent on the caching of that index in the buffer cache. The amount of caching depends on factors, such as the load on the system and the block access patterns of different users, that the optimizer cannot predict. The user can modify the optimizer's assumptions about index caching for nested loops joins by setting this parameter to a value between 0 and 100 percent and thereby indicate what percentage of the index blocks should be assumed to be in the cache. Setting this parameter to a higher value makes nested loops join look less expensive to the optimizer and it will be more likely to pick nested loops joins over hash or sort-merge joins. The default for this parameter is 0, which gives the old optimizer behavior.

OPTIMIZER_INDEX_COST_ADJ

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	100
Range of values:	1-10000

OPTIMIZER_INDEX_COST_ADJ lets the user tune the optimizer behavior for access path selection to be more or less index-friendly. Some users are using the first_rows optimizer mode to get plans that use more index access paths and that are more similar to those generated by the rule-based optimizer. However, the first_rows mode was never intended as a pure mechanism to force the use of indexes. The OPTIMIZER_INDEX_COST_ADJ parameter lets the user adjust the costing of index access paths in the cost-based optimizer and thereby 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, which makes the optimizer cost index access paths a the regular cost. Any other value will make the optimizer cost the access path at that percentage of the regular cost, e.g., setting it to 50 percent, will make the index access path look half as expensive as normal. The legal range of values for this parameter is 1 to 10000 percent. This parameter can be used to tune the performance of a system where it is felt that the optimizer chooses too few or too many index access paths. The adjustment does not apply to user-defined cost functions for domain indexes.

OPTIMIZER_MAX_PERMUTATIONS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	80,000
Range of values:	4-2^32 (~4.3 billion)

OPTIMIZER_MAX_PERMUTATIONS lets the user limit the amount of work the optimizer spends on optimizing queries with large joins. By restricting the number of permutations of the tables the optimizer will consider, the user can ensure that the parse time for the query stays within acceptable limits.

However, in doing so, there is a slight risk that the optimizer will overlook a good plan it would otherwise have found. The default value for this parameter is 80000, which corresponds to the old behavior. Setting this parameter to a value less than 1000 should ensure parse times of a few seconds or less.

OPTIMIZER_MODE

Parameter type:	integer
Parameter class:	dynamic, scope=ALTER SESSION
Default value:	CHOOSE
Range of values:	RULE / CHOOSE / FIRST_ROWS / ALL_ROWS

OPTIMIZER_MODE specifies the behavior of the optimizer. When set to RULE, this parameter causes rule-based optimization to be used unless hints are specified in the query. When set to CHOOSE, the optimizer uses the cost-based approach for a SQL statement if there are statistics in the dictionary for at least one table accessed in the statement. (Otherwise, the rule-based approach is used.)

You can set the goal for cost-based optimization by setting this parameter to FIRST_ROWS or ALL_ROWS. FIRST_ROWS causes the optimizer to choose execution plans that minimize response time. ALL_ROWS causes the optimizer to choose execution plans that minimize total execution time.

For more information about tuning SQL statements, see *Oracle8i Tuning.* For more information about the optimizer, see *Oracle8i Concepts* and *Oracle8i Tuning*.

OPTIMIZER_PERCENT_PARALLEL

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	100
Range of values:	0 - 100

OPTIMIZER_PERCENT_PARALLEL specifies the amount of parallelism that the optimizer uses in its cost functions. The default of 0 means that the optimizer chooses the best serial plan.

A value of 100 means that the optimizer uses each object's degree of parallelism in computing the cost of a full table scan operation. Low values favor indexes, and high values favor table scans.

Cost-based optimization will always be used for any query that references an object with a nonzero degree of parallelism. For such queries, a RULE hint or optimizer mode or goal will be ignored. Use of a FIRST_ROWS hint or optimizer mode will override a non-zero setting of OPTIMIZER_PERCENT_PARALLEL.

OPTIMIZER_SEARCH_LIMIT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	5

OPTIMIZER_SEARCH_LIMIT specifies the maximum number of tables in a query block for which the optimizer will consider join orders with Cartesian products.

If there are more tables than the limit, the optimizer will, if possible, only consider join orderings where all the join are between tables that are connected by equality join conditions. This prevent the optimizer from spending an inordinate amount of time on join orderings that are likely to be suboptimal.

ORACLE_TRACE_COLLECTION_NAME

Parameter type:	string
Parameter class:	static
Default value:	NULL
Range of values:	valid collection name up to 16 characters long (except for platforms that enforce 8-character file names)

ORACLE_TRACE_COLLECTION_NAME specifies the Oracle Trace collection name. This parameter is also used in the output file names (collection definition file .cdf and data collection file .dat). If the ORACLE_TRACE_ENABLE parameter is set 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.

ORACLE_TRACE_COLLECTION_PATH

Parameter type:	string
Parameter class:	static
Default value:	operating system-specific
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**.

ORACLE_TRACE_COLLECTION_SIZE

Parameter type:	integer
Parameter class:	static
Default value:	5242880
Range of values:	0 - 4294967295

ORACLE_TRACE_COLLECTION_SIZE specifies the maximum size, in bytes, of the Oracle Trace collection file. Once the collection file reaches this maximum, the collection is disabled. When the Range of Values is zero, there is no size limit.

ORACLE_TRACE_ENABLE

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

To enable Oracle Trace collections for the server, ORACLE_TRACE_ENABLE should be set and left at TRUE. When set to TRUE, this alone does not start an Oracle Trace collection, but it allows Oracle Trace to be used for that server.

When set to TRUE, an Oracle Trace collection can then be started by using the Oracle Trace Manager application (supplied with the Oracle Diagnostics Pack), or by including a name in the ORACLE_TRACE_COLLECTION_NAME parameter (default = null).

ORACLE_TRACE_FACILITY_NAME

Parameter type:	string
Parameter class:	static
Default value:	oracled
Range of values:	valid facility name up to 16 characters long

ORACLE_TRACE_FACILITY_NAME specifies the Oracle Trace product definition file (.fdf file). The file must be located in the directory pointed to by the ORACLE_ TRACE_FACILITY_PATH parameter. The product definition file contains definition information for all the events and data items which can be collected for a product that used the Oracle Trace data collection API. Products can have multiple product definition files (multiple event sets and data items). The Oracle server has multiple event sets and therefore multiple product definition files. Oracle Corporation recommends that you use the "default" event set for Oracle server collections ORCLED.FDF. See the Oracle Trace documentation for more information on the Oracle server event sets.

ORACLE_TRACE_FACILITY_PATH

Parameter type:	string
Parameter class:	static
Default value:	operating system-specific
Range of values:	full directory pathname

ORACLE_TRACE_FACILITY_PATH specifies the directory pathname where Oracle TRACE facility definition files are located. For Solaris, the file specification is ?/otrace/admin/fdf/. For NT, the file specification is: %OTRACE80%\ADMIN\FDF\.

OS_AUTHENT_PREFIX

Parameter type:	string
Parameter class:	static
Default value:	operating system-specific (typically "OPS$")

OS_AUTHENT_PREFIX authenticates users attempting to connect to the server with the users' operating system account name and password. The value of this parameter is concatenated to the beginning of every user's operating system account. The prefixed username is compared with the Oracle usernames in the database when a connection request is attempted. 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 with some operating systems.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation.

OS_ROLES

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

If OS_ROLES is set to TRUE, the database allows the operating system to identify each username's roles. 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.

If OS_ROLES is set to TRUE, the operating system completely manages the role grants for all database usernames. Any revokes of roles granted by the operating system are ignored, and any previously granted roles are ignored.

The default value, FALSE, causes roles to be identified and managed by the database.

For more information, see the *Oracle8i Administrator's Guide*.

PARALLEL_ADAPTIVE_MULTI_USER

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	If PARALLEL_AUTOMATIC_TUNING = TRUE, then TRUE, else FALSE
Range of values:	TRUE, FALSE

PARALLEL_ADAPTIVE_MULTI_USER, when set to TRUE, enables an adaptive algorithm designed to improve performance in multi-user environments that use parallel execution. It does this by automatically reducing 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 reduction factor is computed using the load on the system and the number of active parallel execution users. This load is compared to a target optimal system load which is computed using the number of CPUs on the system and the parameter PARALLEL_THREADS_PER_CPU among others.

The algorithm assumes that the system has been tuned for optimal performance in a single user environment, and that it works best when the parameter PARALLEL_AUTOMATIC_TUNING set to true, and the tables and hint use the DEFAULT degree of parallelism.

On a 16 CPU machine, for example, the default degree of parallelism could be set to 32. If one user issues a parallel execution, that user gets a degree of 32, effectively using all of the CPUs and memory in the system. When a second user enters the system, that user could get a degree of 16. As the number of users on the system increases, this algorithm will continue to reduce the degree until the users are running using degree 1, when the load on the system significantly exceeds the target optimal load. Over time, the behavior of the system as a whole will be improved.

For more information, see *Oracle8i Tuning*.

PARALLEL_AUTOMATIC_TUNING

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

Note: This parameter applies to Parallel Execution, not the Oracle8i Parallel Server Option.

Enable PARALLEL_AUTOMATIC_TUNING when you want Oracle to determine the default values for parameters that control Parallel Execution. In addition to setting this parameter, you must enable PARALLEL, for the target tables in the system. All subsequent tuning will be done by the system.

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 be allocated from the Large Pool, and will be computed automatically if LARGE_POOL_SIZE is left unset.

This will include setting the PARALLEL_ADAPTIVE_MULTI_USER parameter which will override user-provided hints in favor of maintaining the load on the system within acceptable ranges. The database administrator can override any of the system-provided defaults if desired.

For more information see *Oracle8i Tuning*.

PARALLEL_BROADCAST_ENABLED

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	FALSE
Range of values:	TRUE, FALSE

Note: This parameter refers to Parallel Execution, not the Oracle8*i* Parallel Server Option.

PARALLEL_BROADCAST_ENABLED allows you to improve performance in certain cases involving hash and merge joins. When set to TRUE, if you are joining a very large join result set with a very small result set (size being measured in bytes, rather than number of rows), the optimizer has the option of broadcasting the row sources of the small result set, such that a single table queue will send all of the small set's rows to each of the parallel servers which are processing the rows of the larger set. The result is enhanced performance.

For more information, see *Oracle8i Tuning*.

PARALLEL_DEFAULT_MAX_INSTANCES

Obsoleted in 8.1.3.

PARALLEL_EXECUTION_MESSAGE_SIZE

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent. (Usually 2148 if PARALLEL_AUTOMATIC_TUNING is FALSE, 4096 if PARALLEL_AUTOMATIC_TUNING is TRUE.
Range of values:	2148 - infinity
Multiple instances:	must have the same value

PARALLEL_EXECUTION_MESSAGE_SIZE specifies the size of messages for parallel execution (formerly parallel query, PDML, Parallel Recovery, replication). The default value should be adequate for most applications.

Typical values are 2048 or 4096 bytes. Larger values would require a larger shared pool. 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 will generally be higher.

PARALLEL_INSTANCE_GROUP

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	group consisting of all instances currently active
Range of values:	a string representing a group name

PARALLEL_INSTANCE_GROUP is a Parallel Server parameter and can be used in parallel mode only. This parameter identifies the parallel instance group to be used for spawning parallel execution slaves. Parallel operations will spawn parallel execution slaves only on instances that specify a matching group in their INSTANCE_GROUPS parameter.

If the value assigned to PARALLEL_INSTANCE_GROUP is the name of a group that does not exist, then the operation runs serially. No parallelism is used.

For more information see *Oracle8i Parallel Server Concepts and Administration*.

PARALLEL_MAX_SERVERS

Parameter type:	integer
Parameter class:	static
Default value:	depends on CPU_COUNT, PARALLEL_AUTOMATIC_TUNING, and PARALLEL_ADAPTIVE_MULTI_USER
Range of values:	0 - 256
Multiple instances:	each instance must have either a value of zero or the same value as the other instances

Note: This parameter applies to Parallel Execution, not the Oracle8*i* Parallel Server Option.

PARALLEL_MAX_SERVERS specifies the maximum number of parallel execution servers or parallel recovery processes for an instance. Oracle will increase the number of query servers as demand requires from the number created at instance startup up to this value. The same value should be used for all instances in a parallel server environment.

PARALLEL_MAX_SERVERS will be used to size the Large Pool and other memory structures and parameters that are used by parallel execution.

Proper setting of the PARALLEL_MAX_SERVERS parameter ensures that the number of query servers in use will not cause a memory resource shortage during periods of peak database use.

If PARALLEL_MAX_SERVERS is set too low, some queries may not have a query server available to them during query processing.

Setting PARALLEL_MAX_SERVERS too high leads to memory resource shortages during peak periods, which can degrade performance. For each instance to which you do not want to apply the parallel execution option, set this initialization parameter to zero.

If you have reached the limit of PARALLEL_MAX_SERVERS on an instance and you attempt to query a GV$ view, one additional parallel server process will be spawned for this purpose.

Note that if PARALLEL_MAX_SERVERS is set to zero for an instance, then no additional parallel server process will be allocated to accommodate a GV$ query.

For more information, see *Oracle8i Tuning*.

PARALLEL_MIN_MESSAGE_POOL

Obsoleted in 8.1.3.

PARALLEL_MIN_PERCENT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	0
Range of values:	0 - 100
OK to change:	yes
Multiple instances:	can have different values; application dependent

PARALLEL_MIN_PERCENT specifies the minimum percent of threads required for parallel execution.

Setting this parameter ensures that parallel execution will not be executed sequentially if adequate resources are not available. The default value of 0 means that this parameter is not used.

If too few query slaves are available, an error message is displayed and the query is not executed. Consider the following settings:

```
PARALLEL_MIN_PERCENT = 50
PARALLEL_MIN_SERVERS = 5
PARALLEL_MAX_SERVERS = 10
```

In a system with 20 instances up and running, the system would have a maximum of 200 query slaves available. If 190 slaves are already in use and a new user wants to run a query with 40 slaves (for example, degree 2 instances 20), an error message would be returned because 20 instances (that is, 50% of 40) are not available.

The parameter PARALLEL_MIN_PERCENT can also be used in conjunction with PARALLEL_ADAPTIVE_MULTI_USER. In a multi-user environment, a user or application can set PARALLEL_MIN_PERCENT to a minimum value until sufficient resources are available on the system and an acceptable degree is returned.

For more information see *Oracle8i Tuning*.

PARALLEL_MIN_SERVERS

Parameter type:	integer
Parameter class:	static
Default value:	0
Range of values:	0 - PARALLEL_MAX_SERVERS
Multiple instances:	can have different values

Note: This parameter applies to Parallel Execution, not the Oracle8*i* Parallel Server Option.

PARALLEL_MIN_SERVERS specifies the minimum number of query server processes for an instance. This is also the number of query server processes Oracle creates when the instance is started.

For more information, see *Oracle8i Tuning*.

PARALLEL_SERVER

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE
Multiple instances:	must have the same value

Set PARALLEL_SERVER to TRUE to enable the Parallel Server option.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

PARALLEL_SERVER_IDLE_TIME

Obsoleted in 8.1.3.

PARALLEL_SERVER_INSTANCES

Parameter type:	integer
Parameter class:	static
Default value:	1
Range of values:	any non-zero value

PARALLEL_SERVER_INSTANCES shows the number of instances currently configured. It is used to size SGA structures which depend on the number of instances configured. Setting this parameter properly will improve memory use of the SGA. The default values of several parameters are computed using this number.

PARALLEL_THREADS_PER_CPU

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	OS-dependent, usually 2
Range of values:	any non-zero number

Note: This parameter applies to Parallel Execution, not the Oracle8*i* Parallel Server Option.

PARALLEL_ THREADS_PER_CPU is used to set the default degree of parallelism, and to tune the parallel adaptive and load balancing algorithms. The parameter describes the number of processes or threads that a CPU can handle during parallel execution. The default is platform-dependent. The default provided by the system should be adequate for most cases. This number should be decreased from the default provided if the machine appears to be overloaded when a representative query is executed. The value for this parameter should be increased if the system is I/O bound.

For more information, see *Oracle8i Tuning*.

PARALLEL_TRANSACTION_RESOURCE_TIMEOUT

Obsoleted in 8.1.3.

PARTITION_VIEW_ENABLED

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	FALSE
Range of values:	TRUE, FALSE
OK to change:	yes

If PARTITION_VIEW_ENABLED is set to TRUE, the optimizer prunes (or skips) unnecessary table accesses in a partition view. This parameter also changes the way the cost-based optimizer computes statistics on a partition view from statistics on underlying tables.

PLSQL_V2_COMPATIBILITY

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	FALSE
Range of values:	TRUE, FALSE

The PLSQL_V2_COMPATIBILITY initialization parameter is optional and sets the compatibility level for PL/SQL. The default value is FALSE: PL/SQL V3 behavior is enforced and V2 behavior is not allowed.

If PLSQL_V2_COMPATIBILITY=TRUE, then the following PL/SQL V2 behaviors are accepted when you are running PL/SQL V3:

- PL/SQL will allow elements of an index table passed in as an IN parameter to be modified or deleted. For example,

  ```
  function foo (x IN table_t) is
  begin
  x.delete(2);
  end;
  ```

  In contrast, PL/SQL V3 correctly enforces the read-only semantics of IN parameters and does not let index table methods modify index tables passed in as IN parameters.

- The PL/SQL compiler will allow OUT parameters to be used in expression contexts in some cases (for example, in dot-qualified names on the right-hand side of assignment statements). This behavior is restricted to a few cases - fields of OUT parameters that are record, and OUT parameters referenced in the FROM list of a SELECT statement.

  In contrast, PL/SQL V3 does not permit OUT parameters to be used in expression contexts.

- PL/SQL will allow OUT parameters in the FROM clause of a SELECT list, where their value is read.

- PL/SQL will not return an error on the illegal syntax

  ```
  return expression
  ```

which should be

```
return type
```

In contrast, PL/SQL V3 returns an error.

- PL/SQL will allow the passing of an IN argument into another procedure as an OUT. (This is restricted to fields of IN parameters that are records.)

  In contrast, PL/SQL V3 does not allow the passing of an IN argument into another procedure as an OUT.

- PL/SQL will allow a type to be referenced earlier than its definition in the source.

  In contrast, PL/SQL V3 requires a type definition to precede its use.

PRE_PAGE_SGA

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	FALSE, TRUE
OK to change:	no

PRE_PAGE_SGA is a platform-specific parameter; see your platform-specific documentation for further details.

If PRE_PAGE_SGA is set to TRUE, this parameter touches all the SGA pages, causing them to be brought into memory. As a result, it increases instance startup time and user login time, but it can reduce the number of page faults that occur shortly thereafter. The reduction in page faults allows the instance to reach its maximum performance capability quickly rather than through an incremental buildup. It is most useful on systems that have sufficient memory to hold all the SGA pages without degrading performance in other areas.

PROCESSES

Parameter type:	integer
Parameter class:	static
Default value:	depends on PARALLEL_MAX_SERVERS
Range of values:	6 to operating system-dependent
Multiple instances:	can have different values

For a multiple-process operation, PROCESSES specifies the maximum number of operating system user processes that can simultaneously connect to an Oracle Server. This value should allow for all background processes such as LCK processes, Job Queue processes, and parallel execution processes.

The default values of SESSIONS is derived from PROCESSES. If you alter the value of PROCESSES, you may want to adjust the values of this derived parameter.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the range of values.

PUSH_JOIN_PREDICATE

Obsoleted in 8.1.3.

QUERY_REWRITE_ENABLED

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	FALSE
Range of values:	TRUE, FALSE
Multiple instances:	can have different values
OK to change:	yes

QUERY_REWRITE_ENABLED allows you to enable or disable query rewriting.

Query rewriting is enabled for a particular materialized view only if both the session parameter and the individual materialized view are enabled and when cost-based optimization is enabled.

QUERY_REWRITE_INTEGRITY

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	ENFORCE
Range of values:	ENFORCE, NO_ENFORCE, USE_STALE
Multiple instances:	can have different values
OK to change:	yes

QUERY_REWRITE_INTEGRITY determines the degree to which query rewriting must be enforced by the Oracle server. In the safest level, query rewrite transformations that rely on unenforced relationships are not used.

With ENFORCE, consistency and integrity are enforced and guaranteed by Oracle. With NO_ENFORCE, rewrites are allowed using relationships that have been declared, but that are not enforced by Oracle. With USE_STALE, rewrites are allowed using unenforced relationships, and 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
Parameter class:	static
Default value:	none
Range of values:	all X.500 Distinguished Name format values

Note: Global user functionality is currently available only to beta customers. This feature will be available to all users in a subsequent release of Oracle8i.

This parameter value is the Distinguished Name of the RDBMS server. It is used for retrieving Enterprise Roles from an enterprise directory service. See the *Oracle Advanced Security Administrator's Guide* for more information.

READ_ONLY_OPEN_DELAYED

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

READ_ONLY_OPEN_DELAYED, when set to TRUE, causes datafiles in read-only tablespaces to be accessed for the first time only when an attempt is made to read data stored within them. When set to FALSE, datafiles are accessed at database open time.

This parameter is used to speed certain operations, primarily the ’open database’ operation, for very large databases, when substantial portions of the database are stored in read-only tablespaces. It should be considered for such databases, especially if portions of the read-only data are stored on slow-access devices or hierarchical storage.

Use of this parameter has the following side-effects:

- A missing or bad read-only file will not be detected at open time. It will only be discovered when there is an attempt to access it.
- ALTER DATABASE CHECK DATAFILES will not check read-only files.
- ALTER TABLESPACE <name> ONLINE and ALTER DATABASE DATAFILE <name> ONLINE will not check read-only files. They will be checked only upon the first access.
- V$RECOVER_FILE, V$BACKUP, and V$DATAFILE_HEADER will not access read-only files; read-only files will be indicated in the results list with the error "DELAYED OPEN", with zeroes for the values of other columns.
- V$DATAFILE will not access read-only files; read-only files will have a size of ’0’ listed.
- V$RECOVER_LOG will not access read-only files; logs they may need for recovery will not be added to the list.
- ALTER DATABASE NOARCHIVELOG will not access read-only files; it will proceed even if there is a read-only file that needs recovery.

Note that RECOVER DATABASE and ALTER DATABASE OPEN RESETLOGS will continue to access all read-only datafiles regardless of the parameter value. If you want to avoid accessing read-only files for these operations, those files should be taken offline.

Also note that if a backup controlfile is used, the read-only status of some files may be inaccurate. This may cause some of these operations to return unexpected results. Care should be taken in this situation.

RECOVERY_PARALLELISM

Parameter type:	integer
Parameter class:	static
Default value:	operating system-dependent
Range of values:	operating system-dependent, but cannot exceed PARALLEL_ MAX_SERVERS

RECOVERY_PARALLELISM specifies the number of processes to participate in instance or media recovery. A value of zero or one indicates that recovery is to be performed serially by one process.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

REDUCE_ALARM

Obsoleted in 8.1.3.

REMOTE_DEPENDENCIES_MODE

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM
Default value:	TIMESTAMP
Range of values:	TIMESTAMP / SIGNATURE

REMOTE_DEPENDENCIES_MODE is used with PL/SQL stored procedures. It specifies how dependencies upon remote stored procedures are to be handled by the database.

If this parameter is set to TIMESTAMP, which is the default setting, 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.

If the parameter is set to SIGNATURE, the procedure is allowed to execute as long as the signatures are considered safe. This allows client PL/SQL applications to be run without recompilation.

REMOTE_LOGIN_PASSWORDFILE

Parameter type:	string
Parameter class:	static
Default value:	NONE
Range of values:	NONE/SHARED/EXCLUSIVE
Multiple instances:	should have the same value

REMOTE_LOGIN_PASSWORDFILE specifies whether Oracle checks for a password file and how many databases can use the password file. Setting the parameter to NONE signifies that Oracle should ignore any password file (and, therefore, privileged users must be authenticated by the operating system). Setting the parameter to EXCLUSIVE (*required for the Oracle Parallel Server*) signifies that the password file can be used by only one database and the password file can contain names other than SYS and INTERNAL. Setting the parameter to SHARED allows more than one database to use a password file. However, the only users recognized by the password file are SYS and INTERNAL.

For more information about secure connections for privileged users, see the *Oracle8i Administrator's Guide*.

REMOTE_OS_AUTHENT

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

Setting REMOTE_OS_AUTHENT to TRUE allows authentication of remote clients with the value of OS_AUTHENT_PREFIX.

For more information, see the *Oracle8i Administrator's Guide*.

REMOTE_OS_ROLES

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

Setting REMOTE_OS_ROLES to TRUE allows operating system roles for remote clients. The default value, FALSE, causes roles to be identified and managed by the database for remote clients.

For more information, see the *Oracle8i Administrator's Guide*.

REPLICATION_DEPENDENCY_TRACKING

Parameter type:	boolean
Parameter class:	static
Default value:	TRUE
Range of values:	TRUE, FALSE

Setting REPLICATION_DEPENDENCY_TRACKING to TRUE turns on dependency tracking for read / write operations to the database.

Dependency tracking is essential for the Replication Server to propagate changes in parallel. This is the default value. FALSE allows read / write operations to the database to run faster, but does not produce dependency information for the Replication Server to perform parallel propagations. Users should not specify this value unless they are sure that their application will perform absolutely no read / write operations to replicated tables.

RESOURCE_LIMIT

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	FALSE
Range of values:	TRUE, FALSE

The value of RESOURCE_LIMIT changes the enforcement status of resource limits set in database profiles. A value of FALSE disables the enforcement of resource limits. A value of TRUE enables the enforcement of resource limits.

For more information, see the *Oracle8i Administrator's Guide*.

RESOURCE_MANAGER_PLAN

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	NULL
Range of values:	any valid character string

This parameter dictates which top plan to use for this instance. The resource manager will load this top plan as well as all its descendants (subplans, directives and consumer groups). If the parameter is not specified, the resource manager is, by default, off.

The administrator may use the ALTER SYSTEM command on the parameter to turn on the resource manager (if it was previously off), turn off the resource manager or change the current plan schema (if it was previously on). If a plan is specified that does not exist in the data dictionary, an error message will be returned.

ROLLBACK_SEGMENTS

Parameter type:	string
Parameter class:	static
Default value:	NULL (the instance uses public rollback segments by default if you do not specify this parameter)
Range of values:	any rollback segment names listed in DBA_ROLLBACK_ SEGS except SYSTEM
Multiple instances:	must have different values (different instances cannot specify the same rollback segment)
OK to change:	yes

ROLLBACK_SEGMENTS specifies one or more rollback segments to allocate by name to this instance.

If ROLLBACK_SEGMENTS is set, an 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 from the ratio TRANSACTIONS / TRANSACTIONS_PER_ROLLBACK_SEGMENT).

Note: Never name the SYSTEM rollback segment as a value for the ROLLBACK_ SEGMENTS parameter.

This parameter has the following syntax:

```
ROLLBACK_SEGMENTS = (rbseg_name [, rbseg_name] ... )
```

Although this parameter usually specifies private rollback segments, it can also specify public rollback segments if they are not already in use.

Different instances in an Oracle Parallel Server cannot name the same rollback segment for any of the ROLLBACK_SEGMENTS. Query the data dictionary view DBA_ROLLBACK_SEGS to find the name, segment ID number, and status of each rollback segment in the database.

For more information, see the *Oracle8i Administrator's Guide*.

ROW_CACHE_CURSORS

Obsoleted in 8.1.3.

ROW_LOCKING

Parameter type:	string
Parameter class:	static
Default value:	ALWAYS
Range of values:	ALWAYS / DEFAULT / INTENT
Multiple instances:	must have the same value

ROW_LOCKING specifies whether row locks are acquired when a table is updated or on update. The default of ALWAYS means that only row locks are acquired when a table is updated. DEFAULT is the same as ALWAYS. INTENT means that only row locks are used on a SELECT FOR UPDATE, but at update time table locks are acquired.

For information about tuning SQL statements, see *Oracle8i Tuning*.

SEQUENCE_CACHE_ENTRIES

Obsoleted in 8.1.3.

SEQUENCE_CACHE_HASH_BUCKETS

Obsoleted in 8.1.3.

SERIAL _REUSE

Parameter type:	string LIST
Parameter class:	static
Default Value:	NULL
Range of values:	DISABLE / SELECT / DML / PLSQL / ALL / NULL

This parameter indicates which types of SQL cursors should make use of the serial-reusable memory feature. This feature moves well-structured private cursor memory into the SGA (shared pool) so that it can be reused by sessions executing the same cursor.

If CURSOR_SPACE_FOR_TIME is TRUE, then the value of SERIAL_REUSE is ignored and treated as if it were set to DISABLE. The default NULL value is equivalent to setting the value to DISABLE. Values include:

Table 1–6 Values for the SERIAL_REUSE Initialization Parameter

Value	Description
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.

SERVICE_NAMES

Parameter type:	string
Parameter class:	static
Default value:	DB_NAME.DB_DOMAIN if defined
Range of values:	any ASCII string, or comma-separated list of string names

SERVICE_NAMES specifies the service names supported by the instance.

SERVICE_NAMES is one or more strings which represent the names of the database on the network. Net8 8.1 wants a service name, rather than a SID, to identify a database. It is possible to provide multiple services names so that different usages of a single database can be identified separately. Service names can also be used to identify a single service that is available from two different databases through the use of replication.

Example:

```
SERVICE_NAMES = sales.acme.com, widgetsales.acme.com
```

This value / values will be sent to the TNS Listeners as the service name(s) that this instance belongs to. If the names in this parameter are not domain qualified, they will be qualified with the value of the DB_DOMAIN parameter before being sent to the listener. If this parameter is not specified, as of 8.1, the default value that will be registered is <db_name>.<db_domain>.

Using the above example, the client CONNECT_DATA should read as follows:

```
(CONNECT_DATA=(SERVICE_NAMES=widgetsales.acme.com))
```

See *Net8 Administrator's Guide*.for further information.

SESSION_CACHED_CURSORS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION
Default:	0
Range of values:	0 to operating system-dependent
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 need not reopen the cursor. The value of this parameter is the maximum number of session cursors to keep in the session cursor cache.

SESSION_MAX_OPEN_FILES

Parameter type:	integer
Parameter class:	static
Default value:	10
Range of values:	1 - the least of (50, MAX_OPEN_FILES defined at the OS level)

SESSION_MAX_OPEN_FILES specifies the maximum number of BFILEs that can be opened in any given session. Once this number is reached, subsequent attempts to open more files in the session using DBMS_LOB.FILEOPEN() or OCILobFileOpen() will fail. This parameter is also dependent on the equivalent parameter defined for the underlying operating system.

SESSIONS

Parameter type:	integer
Parameter class:	static
Default value:	derived (1.1 * PROCESSES + 5)

SESSIONS specifies the total number of user and system sessions. The default number is greater than PROCESSES to allow for recursive sessions.

The default values of ENQUEUE_RESOURCES and TRANSACTIONS are derived from SESSIONS. If you alter the value of SESSIONS, you might want to adjust the values of ENQUEUE_RESOURCES and TRANSACTIONS.

With the multi-threaded server, you should adjust the value of SESSIONS to approximately 1.1 * (*total number of connections*).

For more information on memory structures and processes, see *Oracle8i Concepts*.

SHADOW_CORE_DUMP

Parameter type:	string
Parameter class:	static
Default value:	PARTIAL
Range of values:	FULL / PARTIAL

The value of SHADOW_CORE_DUMP determines whether the SGA will be included in core dumps. By default (FULL), the SGA is included in the core dump. If SHADOW_CORE_DUMP=PARTIAL, the SGA is not dumped.

SHARED_MEMORY_ADDRESS

Parameter type:	integer
Parameter class:	static
Default value:	0

SHARED_MEMORY_ADDRESS and HI_SHARED_MEMORY_ADDRESS specify the SGA's starting address at runtime. Many platforms specify the SGA's starting address at linktime; these parameters are ignored on those 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.

SHARED_POOL_RESERVED_MIN_ALLOC

Obsoleted in 8.1.3.

SHARED_POOL_RESERVED_SIZE

Parameter type:	string
Parameter class:	static
Default value:	5% of the value of SHARED_POOL_SIZE
Range of values:	from SHARED_POOL_RESERVED_MIN_ALLOC to one half of SHARED_POOL_SIZE (in bytes)

SHARED_POOL_RESERVED_SIZE specifies the shared pool space which is reserved for large contiguous requests for shared pool memory. This parameter, along with the SHARED_POOL_RESERVED_MIN_ALLOC parameter, can be used to avoid performance degradation in the shared pool from situations where pool fragmentation forces Oracle to search for and free chunks of unused pool to satisfy the current request.

The shared pool contains the library cache of shared SQL requests, the dictionary cache, stored procedures, and other cache structures that are specific to a particular instance configuration. For example, in an MTS configuration, the session and private SQL area for each client process is included in the shared pool.

When the instance is configured for parallel execution, the shared pool includes the parallel execution message buffers.

Proper sizing of the shared pool can reduce resource consumption in at least three ways:

- Parse time is avoided if the SQL statement is already in the shared pool. This saves CPU resources.
- Application memory overhead is reduced, since all applications use the same pool of shared SQL statements and dictionary resources.
- I/O resources are saved, since dictionary elements which are in the shared pool do not require disk access.

Default value for SHARED_POOL_RESERVED_SIZE is 5% of the SHARED_POOL_SIZE. This means that, by default, the reserved list will always be configured.

If SHARED_POOL_RESERVED_SIZE > 1/2 SHARED_POOL_SIZE, Oracle will signal an error.

Ideally, this parameter should be large enough to satisfy any request scanning for memory on the reserved list without flushing objects from the shared pool. The amount of operating system memory, however, may constrain the size of the shared pool. In general, you should set SHARED_POOL_RESERVED_SIZE to 10% of SHARED_POOL_SIZE. For most systems, this value will be sufficient if you have already tuned the shared pool.

SHARED_POOL_RESERVED_SIZE can accept a numerical value or a number followed by the suffix "K" or "M" where "K" means "multiply by 1000" and "M" means "multiply by 1000000".

For more information on this parameter see *Oracle8i Tuning*.

SHARED_POOL_SIZE

Parameter type:	string
Parameter class:	static
Default value:	If 64 bit, 64MB, else 16MB
Range of values:	300 Kbytes - operating system-dependent

SHARED_POOL_SIZE specifies the size of the shared pool in bytes. The shared pool contains shared cursors, stored procedures, control structures, Parallel Execution message buffers among others. Larger values improve performance in multi-user systems. Smaller values use less memory.

If you set PARALLEL_AUTOMATIC_TUNING to TRUE, Parallel Execution message buffers are allocated out of the large pool. If you set it to FALSE, Oracle allocates these buffers from the shared pool. See "LARGE_POOL_SIZE" and *Oracle8i Migration* for details. To monitor utilization of the shared pool, query the view V$SGASTAT. For example:

```
select * from v$sgastat;
```

SHARED_POOL_SIZE can accept a numerical value or a number followed by the suffix "K" or "M" where "K" means "multiply by 1000" and "M" means "multiply by 1000000".

For more information, see the *Oracle8i Administrator's Guide*.

SORT_AREA_RETAINED_SIZE

Parameter type:	integer
Parameter class:	dynamic, scope= ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	the value of SORT_AREA_SIZE
Range of values:	from the value equivalent of two database blocks to the value of SORT_AREA_SIZE

SORT_AREA_RETAINED_SIZE specifies the maximum amount, in bytes, of User Global Area (UGA) memory retained after a sort run completes. The retained size controls the size of the read buffer which is used 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. Multiple sort spaces of this size may be allocated for each query. Usually, only one or two sorts occur at one time, even for complex queries. In some cases, though, additional concurrent sorts are required. Each sort keeps its own memory area, as specified by SORT_AREA_RETAINED_SIZE. If the multi-threaded 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.

For more information, see *Oracle8i Concepts*.

SORT_AREA_SIZE

Parameter type:	integer
Parameter class:	dynamic, scope= ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	operating system-dependent
Range of values:	minimum-the value equivalent of six database blocks maximum-system-dependent

SORT_AREA_SIZE specifies the maximum amount, in bytes, of memory to use for a sort. After the sort is complete and all that remains to do is to return the rows, the memory is released down to the size specified by SORT_AREA_RETAINED_SIZE. After the last row is returned, all memory is freed.

Increasing SORT_AREA_SIZE size improves the efficiency of large sorts. Multiple allocations never exist; there is only one memory area of SORT_AREA_SIZE for each user process at any time.

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 usually adequate for most OLTP operations. You might want to adjust this parameter for decision support systems, batch jobs, or large CREATE INDEX operations.

For more information, see *Oracle8i Concepts*. See also your operating system-specific Oracle documentation for the default value on your system.

SORT_DIRECT_WRITES

Obsoleted in 8.1.3.

SORT_MULTIBLOCK_READ_COUNT

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SESSION, ALTER SYSTEM DEFERRED
Default value:	2
Range of values:	1 - system-dependent value

SORT_MULTIBLOCK_READ_COUNT specifies the number of database blocks to read each time a sort performs a read from a temporary segment. Temporary segments are used by a sort when the data does not fit in SORT_ARE_SIZE of memory. In these situations, sort writes out sections of data to temporary segments in the form of sorted runs. Once all the data has been partially sorted to these runs, sort merges the runs by reading pieces of them from the temporary segment into memory to produce the final sorted output. If SORT_AREA_SIZE is not large enough to merge all the runs at once, subsets of the runs are merged in a number of merge passes.

Increasing the SORT_MULTIBLOCK_READ_COUNT parameter forces sort to read a larger section of each run into memory during a merge pass. This reduces the merge width, or number of runs that can be merged in one merge pass, and may increase the number of merge passes. Each merge pass produces an intermediate run on disk, a run that contains all the data that was part of the runs that were just merged. Any increase in I/O throughput obtained by increasing SORT_MULTIBLOCK_READ_COUNT needs to be balanced with a possible increase in total amount of I/O performed due to an increase in the number of merge passes.

Sort may read more blocks at a time than what is specified by SORT_MULTIBLOCK_READ_COUNT in cases where the number of runs, and therefore the merge width is small relative to SORT_AREA_SIZE.

SORT_READ_FAC

Obsoleted in 8.1.3.

SORT_SPACEMAP_SIZE

Obsoleted in 8.1.3.

SORT_WRITE_BUFFER_SIZE

Obsoleted in 8.1.3.

SORT_WRITE_BUFFERS

Obsoleted in 8.1.3.

SPIN_COUNT

Obsoleted in 8.1.3.

SQL92_SECURITY

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

SQL92_SECURITY specifies whether table-level SELECT privileges are required to execute an update or delete that references table column values.

SQL_TRACE

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
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. Because the SQL trace facility causes system overhead, you should run the database with the value TRUE only for the purpose of collecting statistics. The value can also be changed using the DBMS_SYSTEM package.

For more information about performance diagnostic tools, see *Oracle8i Tuning*. See also *Oracle8i SQL Reference*.

STANDBY_ARCHIVE_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	NULL
Range of values:	NULL string or valid path / device name other than raw

STANDBY_ARCHIVE_DEST defines the standby database destination for the archive redo log file group. It is used by the RFS server on the standby database as the archive log destination, so that it can be specified separately from the LOG_ARCHIVE_DEST parameter. Note that there is no corresponding RFS_ARCHIVE_DUPLEX_DEST parameter.

STANDBY_ARCHIVE_DEST specifies the location of archivelogs arriving from a primary instance. STANDBY_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT are used to fabricate the fully-qualified archivelog filename at the standby site.

Use the view V$ARCHIVE_DEST of the standby instance to see the values for this parameter. The syntax is:

```
STANDBY_ARCHIVE_DEST='filespec'
```

STAR_TRANSFORMATION_ENABLED

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SESSION
Default value:	FALSE
Range of values:	TRUE, FALSE, or TEMP_DISABLE

The value of STAR_TRANSFORMATION_ENABLED determines whether a cost-based query transformation will be applied to star queries. If set to TRUE, the optimizer will consider performing a cost-based query transformation on the star query. If set to FALSE, the transformation will not be applied. If set to 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.

For more information, see *Oracle8i Concepts*.

TAPE_ASYNCH_IO

Parameter type:	boolean
Parameter class:	static
Default value:	TRUE
Range of values:	TRUE, FALSE

TAPE_ASYNCH_IO can be used to control whether I/O to sequential devices (for example, BACKUP/RESTORE of Oracle data TO/FROM tape) is asynchronous. If a platform supports asynchronous I/O to sequential devices, it is recommended that this parameter is left to its default. However, if the asynchronous I/O implementation is not stable, TAPE_ASYNCH_IO can be used to disable its use. If a platform does not support asynchronous I/O to sequential devices, this parameter has no effect.

TEMPORARY_TABLE_LOCKS

Obsoleted in 8.1.3.

THREAD

Parameter type:	integer
Parameter class:	static
Default value:	0
Range of values:	0 - maximum number of enabled threads
Multiple instances:	if specified, must have different values

THREAD is applicable only to instances that intend to run in parallel (shared) mode.

THREAD specifies the number of the redo thread that is to be used by the instance. Any available redo thread number can be used, but an instance cannot use the same thread number as another instance.

Also, an instance cannot start when its redo thread is disabled. A value of zero causes an available, enabled public thread to be chosen. An instance cannot mount a database if the thread is used by another instance or if the thread is disabled.

Redo threads are specified with the THREAD option of the ALTER DATABASE ADD LOGFILE command. Redo threads are enabled with the ALTER DATABASE ENABLE [PUBLIC] THREAD command. The PUBLIC keyword signifies that the redo thread may be used by any instance. This is useful when running systems that have faster access to disks from certain nodes.

Thread 1 is the default thread in exclusive mode. An instance running in exclusive mode can specify THREAD to use the redo log files in a thread other than thread 1.

For more information, see *Oracle8i Parallel Server Concepts and Administration* and *Oracle8i SQL Reference*.

TIMED_OS_STATISTICS

Parameter type:	integer
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	0 (not to refresh OS statistics)
Range of values:	time in seconds

TIMED_OS_STATISTICS can be used by the system administrator to gather operating system statistics when a request is made from the client to the server or a request completes. For dedicated servers, OS statistics are gathered at the time of user logon and thereafter when calls are popped, provided the specified time limit has expired. The statistics are also gathered at the user logoff time.

For Mult-threaded Servers, if TIMED_OS_STATISTICS has a nonzero value, then OS statistics are gathered when calls are pushed or popped.

Gathering OS statistics is expensive and should be done on an as-needed basis. Further, as this is a dynamic parameter, this should be set to zero immediately after the need for gathering OS statistics has been satisfied.

TIMED_STATISTICS

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM, ALTER SESSION
Default value:	FALSE
Range of values:	TRUE, FALSE

If TIMED_STATISTICS is FALSE, the statistics related to time are always zero and the server can avoid the overhead of requesting the time from the operating system. To turn on statistics, set the value to TRUE. Normally, TIMED_STATISTICS should be FALSE.

On some systems with very fast timer access, timing might be enabled even when the parameter is set to FALSE. On these systems, setting the parameter to TRUE might produce more accurate statistics for long-running operations.

For more information about performance diagnostic tools, see *Oracle8i Tuning*.

TRANSACTION_AUDITING

Parameter type:	boolean
Parameter class:	dynamic, scope = ALTER SYSTEM DEFERRED
Default value:	TRUE
Range of values:	TRUE, FALSE

If TRANSACTION_AUDITING is TRUE, the transaction layer generates a special redo record which contains session and user information. This information includes the user logon name, user name, the session ID, some operating system information, and client information. On each successive commit, the transaction layer generates a record that contains only the session ID (which links back to the first record since it also contains the session ID). These records might be useful if using a redo log analysis tool.

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

TRANSACTIONS

Parameter type:	integer
Parameter class:	static
Default value:	derived (1.1 * SESSIONS)
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.

For more information about memory structures and processes, see *Oracle8i Concepts* and the *Oracle8i Administrator's Guide*.

TRANSACTIONS_PER_ROLLBACK_SEGMENT

Parameter type:	integer
Parameter class:	static
Default value:	5
Range of values:	1 - operating system-dependent
Multiple instances:	can have different values

TRANSACTIONS_PER_ROLLBACK_SEGMENT specifies the number of concurrent transactions allowed per rollback segment. 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.

More rollback segments can be acquired if they are named in the parameter ROLLBACK_SEGMENTS.

For more information, see the *Oracle8i Administrator's Guide*. See also your operating system-specific Oracle documentation for the range of values.

USE_INDIRECT_DATA_BUFFERS

Parameter type:	boolean
Parameter class:	static
Default value:	FALSE
Range of values:	TRUE, FALSE

USE_INDIRECT_DATA_BUFFERS controls the use of the extended buffer cache mechanism for 32-bit platforms that can support more than 4GB of physical memory. It is ignored on other platforms.

USE_ISM

Obsoleted in 8.1.3.

USER_DUMP_DEST

Parameter type:	string
Parameter class:	dynamic, scope = ALTER SYSTEM
Default value:	operating system-dependent
Range of values:	valid local pathname, 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 to **C:\ORACLE\UTRC** on MS-DOS; to **/oracle/utrc** on UNIX; or to **DISK$UR3:[ORACLE.UTRC]** on VMS.

For more information about performance diagnostic tools, see *Oracle8i Tuning*. See also your operating system-specific Oracle documentation for the range of values.

UTL_FILE_DIR

Parameter type:	string
Parameter class:	static
Default value:	none
Range of values:	any valid directory path

UTL_FILE_DIR allows database administrators to specify directories that are permitted for PL/SQL file I/O. Each directory must be specified with a separate UTL_FILE_DIR parameter in the INIT.ORA file.

Note that all users can read or write all files specified in the UTL_FILE_DIR parameter(s). This means that all PL/SQL users must be trusted with the information in the directories specified by the UTL_FILE_DIR parameters.

2

Static Data Dictionary Views

This chapter contains descriptions of data dictionary tables and views. To see the data dictionary views available to you, query the view DICTIONARY.

See Chapter 3, "Dynamic Performance (V$) Views" for descriptions of the V$ views.

In Trusted Oracle Server, each of the dictionary tables and views contains a column that indicates the label of each row in the table or view. Trusted Oracle also provides some additional dictionary tables and views, and some Oracle dictionary tables and views contain columns that support compatibility with Trusted Oracle applications.

See your Trusted Oracle documentation for more information about Trusted Oracle dictionary tables and views.

Data Dictionary Views

The following is an alphabetical reference of the data dictionary views accessible to all users of an Oracle Server. Most views can be accessed by any user with the CREATE_SESSION privilege.

The data dictionary views that begin with DBA_ are restricted. These views 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.

ALL_ALL_TABLES

This view describes all of the tables (object tables and relational tables) accessible to the user.

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 bloc
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)		Has table been backed up since last modification?

Column	Datatype	NULL	Description
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
TABLE_TYPE_OWNER	VARCHAR2(0)		Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(0)		Type of the table if the table is a typed table
PACKED	VARCHAR2(0)		If the table is a typed table, does it store objects in packed format?
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)		Type of the table if the table is a typed table
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled

Column	Datatype	NULL	Description
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction, else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

ALL_ARGUMENTS

This view lists all of the arguments in the object which are accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username of the owner of the object
OBJECT_NAME	VARCHAR2(30)		Procedure or function name
OVERLOAD	VARCHAR2(40)		Overload unique identifier
PACKAGE_NAME	VARCHAR2(30)		Package name
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
ARGUMENT_NAME	VARCHAR2(30)		Argument name
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: 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)		Owner name of the type

Column	Datatype	NULL	Description
TYPE_NAME	VARCHAR2(30)		Name
TYPE_SUBNAME	VARCHAR2(30)		This is valid only in case of package local types; in such cases, the package name is the name and the type name is the subname
TYPE_LINK	VARCHAR2(128)		Database link valid only in case of package local types, in case the package is remote

ALL_ASSOCIATIONS

This view lists user-defined statistics information.

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	Object name 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)		Schema type of the object - column, type, package or function
STATSTYPE_ SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of Statistics type which contains the cost, selectivity or stats funcs
DEF_SELECTIVITY	NUMBER		Default Selectivity if any of the object
DEF_CPU_COST	NUMBER		Default CPU cost if any of the object
DEF_IO_COST	NUMBER		Default I/O cost if any of the object
DEF_NET_COST	NUMBER		Default Networking cost if any of the object

ALL_CATALOG

This view lists all tables, views, synonyms, and sequences accessible to the user.

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

ALL_CLUSTERS

This view list all clusters accessible to the user.

Column	Datatype	NULL	Description
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
OWNER	VARCHAR2(30)	NOT NULL	The owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	The 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
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 hash 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)		The number of threads per instance for scanning the cluster
INSTANCES	VARCHAR2(10)		The 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
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
SINGLE_TABLE	VARCHAR2(5)		Y if the cluster is single table; N if not

ALL_CLUSTER_HASH_EXPRESSIONS

This view list hash functions for all accessible clusters.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	The name of the cluster
HASH_EXPRESSION	LONG		Text of hash function of cluster

ALL_COL_COMMENTS

This view lists comments on columns of accessible tables and views.

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

This view lists grants on columns for which the user or PUBLIC is the grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		YES if the privileges were granted with ADMIN OPTION; otherwise NO

ALL_COL_PRIVS_MADE

This view lists grants on columns for which the user is owner or grantor.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Username 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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_COL_PRIVS_RECD

This view lists grants on columns for which the user or PUBLIC is the grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NUL	Username 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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_COLL_TYPES

This view displays the named collection types accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
COLL_TYPE	VARCHAR2(30)	NOT NULL	Collection type
UPPER_BOUND	NUMBER		Maximum size of the VARRAY type

Column	Datatype	NULL	Description
ELEM_TYPE_MOD	VARCHAR2(7)		Type modifier of the element
ELEM_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the element
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the type of the element
LENGTH	NUMBER		Length of the CHAR element or maximum length of the VARCHAR or VARCHAR2 element
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL element or binary precision of the FLOAT element
SCALE	NUMBER		Scale of the NUMBER or DECIMAL element
CHARACTER_SET _NAME	VARCHAR2(44)		The name of the character set: CHAR_CS NCHAR_CS
ELEM_STORAGE	VARCHAR2(7)		Storage optimization specification for VARRAY of numeric elements
NULLS_STORED	VARCHAR2(3)		Is null information stored with each VARRAY element?

ALL_CONS_COLUMNS

This view contains information about accessible columns in constraint definitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name associated with table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name associated with column or attribute of the object type column specified in the constraint definition
POSITION	NUMBER		Original position of column or attribute in definition

Note: If you create a constraint on a user-defined REF column, the system creates the constraint on the exploded columns that make up the REF column and hence column names of the constraint will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

ALL_CONSTRAINTS

This view lists constraint definitions on accessible tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with 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), or V (with check option, on a view), or O (with read only, on a view)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name associated with table with constraint definition
SEARCH_CONDITION	LONG		Text of search condition for table check
R_OWNER	VARCHAR2(30)		Owner of table used in referential constraint
R_CONSTRAINT_NAME	VARCHAR2(30)		Name of unique constraint definition for referenced table
DELETE_RULE	VARCHAR2(9)		Delete rule for a referential constraint: CASCADE / NO ACTION
STATUS	VARCHAR2(8)		Enforcement status of constraint: ENABLED or DISABLED
DEFERRABLE	VARCHAR2(14)		Indicates whether the constraint is deferrable
DEFERRED	VARCHAR2(9)		Indicates whether the constraint was initially deferred
VALIDATED	VARCHAR2(13)		Indicates whether all data obeys the constraint: VALIDATED, NOT VALIDATED
GENERATED	VARCHAR2(14)		Indicates whether the name system is generated
BAD	VARCHAR2(3)		Creating this constraint should give ORA-02436. Rewrite it before 2000 AD.
RELY	VARCHAR2(4)		If set, this flag will be used in the optimizer
LAST_CHANGE	DATE		Indicates when the constraint was last enabled or disabled

ALL_CONTEXT

This view lists active context information.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	The name of the context namespace
SCHEMA	VARCHAR2(30)	NOT NULL	The schema name of the designated package which can set attributes using this namespace
PACKAGE	VARCHAR2(30)	NOT NULL	The package name of the designated package which can set attributes using this namespace

ALL_DB_LINKS

This view lists database links accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username 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
HOST	VARCHAR2(200)		Net8 string for connect
CREATED	DATE	NOT NULL	Creation time of the database link

ALL_DEF_AUDIT_OPTS

This view contains default object-auditing options that will be applied when objects are created.

Column	Datatype	NULL	Description
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL / UNSUCCESSFUL
AUD	VARCHAR2(3)		Auditing AUDIT WHENEVER SUCCESSFUL / UNSUCCESSFUL
COM	VARCHAR2(3)		Auditing COMMENT WHENEVER SUCCESSFUL / UNSUCCESSFUL
DEL	VARCHAR2(3)		Auditing DELETE WHENEVER SUCCESSFUL / UNSUCCESSFUL
GRA	VARCHAR2(3)		Auditing GRANT WHENEVER SUCCESSFUL / UNSUCCESSFUL
IND	VARCHAR2(3)		Auditing INDEX WHENEVER SUCCESSFUL / UNSUCCESSFUL
INS	VARCHAR2(3)		Auditing INSERT WHENEVER SUCCESSFUL / UNSUCCESSFUL
LOC	VARCHAR2(3)		Auditing LOCK WHENEVER SUCCESSFUL / UNSUCCESSFUL

Column	Datatype	NULL	Description
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL / UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL / UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REF	VARCHAR2(3)		Auditing REFERENCES WHENEVER SUCCESSFUL / UNSUCCESSFU (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL / UNSUCCESSFUL

ALL_DEPENDENCIES

This view lists dependencies between objects accessible to the user. Dependencies on views created without any database links are also available.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of object
TYPE	VARCHAR2(12)		Type of object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY, TRIGGER
REFERENCED _OWNER	VARCHAR2(30)		Owner of the parent object
REFERENCED _NAME	VARCHAR2(64)		Type of parent object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY, TRIGGER
REFERENCED_TYPE	VARCHAR2(12)		Type of referenced object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY, TRIGGER
REFERENCED_ LINK_NAME	VARCHAR2(128)		Name of the link to the parent object (if remote)
DEPENDENCY_TYPE	VARCHAR2(4)		Two values: REF when the dependency is a REF dependency; HARD otherwise

ALL_DIMENSIONS

This view represents dimension objects.

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)		'Y' if this dimension is in an invalid state, else 'N'

Column	Datatype	NULL	Description
REVISION	NUMBER		Dimension revision level

ALL_DIM_ATTRIBUTES

This view represents the relationship between a dimension level and a functionally dependent column. The table that the level columns are in must match the table of the dependent column.

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

This view represents a 1:n hierarchical relationship between a pair of levels in a dimension.

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

ALL_DIM_HIERARCHIES

This view represents a dimension hierarchy.

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

This view represents a join between two dimension tables. The join is always specified between a parent dimension level column and a child column.

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	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 key column
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column

ALL_DIM_LEVELS

This view represents a dimension level. All columns of a dimension level must come from the same relation.

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_DIM_LEVEL_KEY

This view represents a column of a dimension level. The position of a column within a level is specified by KEY_POSITION.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

ALL_DIRECTORIES

This view contains the description of all directories accessible to the user.

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

This view lists current errors on all objects accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of object
TYPE	VARCHAR2(12)		Type of object: VIEW, PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
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_IND_COLUMNS

This view lists columns of the indexes on accessible tables.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of object type column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column
DESCEND	VARCHAR2(4)		Y/N, Y if this column is sorted in descending order

Note: If you create an index on a user-defined REF column, the system creates the index on the exploded columns that make up the REF column and hence column names of the index will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

ALL_IND_EXPRESSIONS

This view lists functional index expressions on accessible tables.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Functional index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

ALL_IND_PARTITIONS

This view describes, for each index partition, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE that the current user can access.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)		Index owner
INDEX_NAME	VARCHAR2(30)		Index name
COMPOSITE	VARCHAR2(3)		’YES’ if the partition belongs to a Local index on a table partitioned using Composite method; ’NO’ otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER		Length of partition bound value expression
PARTITION_POSITION	NUMBER		Position of the partition within the index
STATUS	VARCHAR2(8)		Indicates whether index partition is usable or not
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
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(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key
AVG_DATA_BLOCKS _PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for

Column	Datatype	NULL	Description
NUM_ROWS	NUMBER		Number of rows returned by the ANALYZE command
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for the partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

ALL_IND_SUBPARTITIONS

This view describes, for each index subpartition that the current user owns, the partition level partitioning information, the storage parameters for the subpartition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	The owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index 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
STATUS	VARCHAR2(8)		Indicates 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

Column	Datatype	NULL	Description
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key
AVG_DATA_BLOCKS _PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for
NUM_ROWS	NUMBER		Number of rows in this index partition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The buffer pool for the partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

ALL_INDEXES

This view contains descriptions of indexes on tables accessible to the user. To gather statistics for this view, use the SQL command ANALYZE. This view supports parallel partitioned index scans.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username of the owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(12)		Type of index
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
UNIQUENESS	VARCHAR2(9)		Uniqueness status of the index: UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(11)		Enabled or disabled
PREFIX_LENGTH	NUMBER		The number of columns in the prefix of the key used for compression

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the index
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent
NEXT_EXTENT	NUMBER		Size of secondary extents
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		User column-id for last column to be included in index organized table top index
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	VARCHAR(2(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. This statistic is 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. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns. This statistic is rounded to the nearest integer.

Column	Datatype	NULL	Description
CLUSTERING_FACTOR	NUMBER		Statistic that represents the amount of order of the rows in the table based on the values of the index. If its value is near the number of blocks, then the table is very well ordered. In such a case, the index entries in a single leaf block tend to point to rows in the same data blocks. If its value is near the number of rows, then the table is very randomly ordered. In such a case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
STATUS	VARCHAR2(8)		State of the index: DIRECT LOAD, VALID or INPROGS (a DDL operation on the domain index is in progress).
DOMIDX_STATUS	VARCHAR		Reflects the status of the domain index. A NULL value means that the specified index is not a domain index. A Value of VALID means that the index is a domain index and the index does not have any errors. If the value of this column is IDXTYP_INVLD it means that the indextype corresponding to this domain index is invalid.
DOMIDX_OPSTATUS	VARCHAR		Reflects the status of an operation that was performed on the domain index. A value of NULL indicates that the specified index is not a domain index. A value of VALID specifies that the index does not have any errors. A value of FAILED indicates that the operation that was performed on the domain index failed with an error.
FUNCIDX_STATUS	VARCHAR		A value of NULL indicates a non function-based index. ENABLED indicates the function-based index is enabled. DISABLED indicates the function-based index is disabled.
NUM_ROWS	NUMBER		Number of rows in this index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze this index
LAST_ANALYZED	DATE		Timestamp for when this index was last analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index, NULL if PARTITIONED=NO.
INSTANCES	VARCHAR2(40)		Number of instances across which the indexes to be scanned. NULL if PARTITIONED=NO.
PARTITIONED	VARCHAR2(3)		Indicates whether this index is partitioned. Set to 'YES' if it is partitioned
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it places in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this index system generated?
SECONDARY	VARCHAR2(1)		Is the index object created as part of icreate for domain indexes?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

Column	Datatype	NULL	Description
DURATION	VARCHAR2(15)		The duration
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

ALL_INDEXTYPES

This view lists all the indextypes.

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
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of indextype implementation
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype

ALL_INDEXTYPE_OPERATORS

This view lists all the operators supported by indextypes.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
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

ALL_JOBS

This view lists all jobs in the database.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change it
LOG_USER	VARCHAR2(30)	NOT NULL	USER who was logged in when the job was submitted

Column	Datatype	NULL	Description
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 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 (usually NULL if not executing)
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2(8))		Same as NEXT_DATE. The job becomes due for execution at this time
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job, in seconds
BROKEN	VARCHAR2(1)		If Y, no 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		How many times has this job 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)		ALTER SESSION parameters describing the NLS environment of the job
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0

ALL_LABELS

This is a Trusted Oracle Server view that lists system labels. For more information, see your Trusted Oracle documentation.

ALL_LIBRARIES

This new data view lists all the libraries that a user can access.

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)		Is the library dynamically loadable? (YES or NO)
STATUS	VARCHAR2(7)		Status of the library

ALL_LOBS

This view displays the LOBs contained in tables accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table containing the LOB
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table containing the LOB
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB segment
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB index
CHUNK	NUMBER		Size of the LOB chunk as a unit of allocation / manipulation in bytes
PCTVERSION	NUMBER	NOT NULL	Maximum percentage of the LOB space used for versioning
CACHE	VARCHAR2(3)		Indicates whether the LOB is accessed through the buffer cache
LOGGING	VARCHAR2(3)		Indicates whether the changes to the LOB are logged
IN_ROW	VARCHAR2(3)		Are some of the LOBs stored with the base row?

ALL_LOB_PARTITIONS

This view displays the LOBs contained in tables accessible to the user.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Table owner
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a partitioned LOB item
PARTITION_NAME	VARCHAR2(30)		Name of a table partition

Column	Datatype	NULL	Description
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the lOB item
COMPOSITE	VARCHAR2(3)		Is it a Composite partition (one of YES, NO)?
CHUNK	NUMBER		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size of the initial extent in bytes for a LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size of secondary extents in bytes for a LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for a LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of a LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

ALL_LOB_SUBPARTITIONS

This view displays partition-level attributes of LOB data subpartitions.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Table owner
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a partitioned LOB item

Column	Datatype	NULL	Description
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition to which this LOB data subpartition belongs
SUBPARTITION_NAME	VARCHAR2(30)		Name of a table subpartition to which this LOB subpartition corresponds
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		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size of the initial extent in bytes for a LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size of secondary extents in bytes for a LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for a LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of a LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

ALL_METHOD_PARAMS

This view is a description view of method parameters of types accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type

Column	Datatype	NULL	Description
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)
PARAM_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number or position
PARAM_MODE	VARCHAR2(6)		Mode of the parameter
PARAM_TYPE_MOD	VARCHAR2(7)		Type modifier of the parameter
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)		The name of the character set: CHAR_CS or NCHAR_CS

ALL_METHOD_RESULTS

This view is a description view of method results of types accessible to the user.

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)
RESULT_TYPE_MOD	VARCHAR2(7)	NOT NULL	Type modifier of the result
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		The name of the character set: CHAR_CS or NCHAR_CS

ALL_MVIEW_AGGREGATES

This view represents the grouping functions (aggregated measures) that appear in the SELECT list of an aggregated materialized view.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view index

Column	Datatype	Null	Description
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Position of this measure within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Set to Y if DISTINCT aggregation
MEASURE	LONG		Contains the SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*)

ALL_MVIEW_ANALYSIS

This view represents the materialized views that potentially support query rewrite and that have additional information that is available for analysis by applications. This view excludes any materialized view that references a remote table or that includes a reference to a non-static value such as SYSDATE or USER. It also excludes any materialized view that was created as a snapshot prior to Oracle8.1 and that was 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
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view table object
CONTAINER_NAME	VARCHAR2(30)		Name of the snapshot that the materialized view data is contained in
LAST_REFRESH_SCN	NUMBER		The System Change Number
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		FORCE, FAST, COMPLETE, or NONE
SUMMARY	VARCHAR2(1)		Y if this materialized view includes a GROUP BY clause or aggregation, else N
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
INCREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
CONTAINS_VIEWS	VARCHAR2(1)		Y if this materialized view contains a view in its definition, else N

Column	Datatype	Null	Description
UNUSABLE	VARCHAR2(1)		Y if this materialized view is in an unusable state (inconsistent data), else N. A materialized view can be in an unusable state if a system failure occurs during a full refresh
RESTRICTED_SYNTAX	VARCHAR2(1)		Y if this materialized view uses restricted materialized view syntax, else N
INC_REFRESHABLE	VARCHAR2(1)		Y if this materialized view can be incrementally refreshed, else N
KNOWN_STALE	VARCHAR2(1)		Y if the data contained in the materialized view is known to be inconsistent with the detail data, else N
INVALID	VARCHAR2(1)		Y if this materialized view is in an invalid state (inconsistent metadata), else N
QUERY_LEN	NUMBER	NOT NULL	The length of the query field
QUERY	LONG	NOT NULL	SELECT expression of the materialized view definition
REVISION	NUMBER		The materialized view metadata revision level

ALL_MVIEW_DETAIL_RELATIONS

This view 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. Inline views in the materialized view definition are not represented in this table.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, MATERIALIZED VIEW, or SNAPSHOT
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

ALL_MVIEW_JOINS

This view represents a join between two columns in the WHERE clause of a materialized view.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or 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_KEYS

This view 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. Inline views in the materialized view definition are not represented in this table.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or 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_NESTED_TABLES

This view describes the nested tables in tables accessible to the user.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)		Owner of the nested table

Column	Datatype	Null	Description
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)		Indication of default or user-specified storage for the varray
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column locator or value

ALL_OBJECT_TABLES

This view contains descriptions of the object tables accessible to the user.

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)	NOT NULL	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

Column	Datatype	NULL	Description
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)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)	NOT NULL	Type of the table if the table is a typed table
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?

Column	Datatype	NULL	Description
SECONDARY	VARCHAR2(1)		Is this table object created as part of icreate for domain indexes?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		The 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)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

ALL_OBJECTS

This view lists objects accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username of the owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the sub-object (for example, partition)
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
DATA_OBJECT_ID	NUMBER		Object number of the segment which contains the object
OBJECT_TYPE	VARCHAR2(15)		Type of the object (eg, 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 creation of the object (character data)
STATUS	VARCHAR2(7)		Status of the object: VALID, INVALID, or N/A
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it placed in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this object system generated?
SECONDARY	VARCHAR2(1)		Is this a secondary object created as part of create for domain indexes?

ALL_OPANCILLARY

This view lists ancillary information for operators.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of primary operator

ALL_OPARGUMENTS

This view lists argument information for operator bindings.

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
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

ALL_OPBINDINGS

This view lists operator bindings.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		Name of the schema of the return type - not null only for ADTs
RETURN_TYPE	VARCHAR2(30)		Name of the return type

Column	Datatype	NULL	Description
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		Schema of the implementation type of the indextype
IMPLEMENTATION_TYPE	VARCHAR2(30)		Implementation type of the indextype

ALL_OPERATORS

This view lists operators.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

ALL_OUTLINE_HINTS

This view lists the set of hints which make up the outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline
NODE	NUMBER		I.D. of the query or subquery to which the hint applies. The top-level query is labelled 1. Subqueries are assigned sequentially numbered labels, starting with 2
JOIN_POS	NUMBER		Position of the table in the join order. The JOIN_POS column is 0 for all hints except the access method hints. The access method hints identify a table to which the hint and the join position apply
HINT	VARCHAR2(512)		Text of the hint

ALL_OUTLINES

This view lists information about outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		The user-specified or generated name of the stored outline. The name must be of a form that can be expressed in SQL

Column	Datatype	NULL	Description
CATEGORY	VARCHAR2(30)		A user-defined name used to group outlines into collections
USED	VARCHAR2(9)		Flag indicating whether the outline has ever been used
TIMESTAMP	DATE		Timestamp at which the outline was created
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

ALL_PART_COL_STATISTICS

This view contains column statistics and histogram information for table partitions that the current user can access.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2(30)		Table partition 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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
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_PART_HISTOGRAMS

This view contains the histogram data (end-points per histogram) for histograms on table partitions that the current user can access.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner name
TABLE_NAME	VARCHAR2(30)		Table name
PARTITION_NAME	VARCHAR2(30)		Table partition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

ALL_PART_INDEXES

This view lists the object level partitioning information for all partitioned indexes that the current user can access.

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)		In addition to RANGE, may specify HASH
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this index
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method, the default number of subpartitions, if any
PARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method with Hash subpartitioning, this column will contain a number of columns in the subpartitioning key
LOCALITY	VARCHAR2(6)		Indicates whether this partitioned index is LOCAL or GLOBAL
ALIGNMENT	VARCHAR2(12)		Indicates whether this partitioned index is PREFIXED or NON-PREFIXED

Column	Datatype	NULL	Description
DEF_TABLESPACE _NAME	VARCHAR2(30)		Default TABLESPACE, for LOCAL index, for ADD / SPLIT TABLE partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, for LOCAL index, for ADD TABLE partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, for LOCAL index, for ADD TABLE partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, for LOCAL index, for ADD TABLE partition
DEF_INITIAL_EXTENT	VARCHAR2(40)	NOT NULL	Default INITIAL, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)
DEF_NEXT_EXTENT	VARCHAR2(40)	NOT NULL	Default NEXT, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	VARCHAR2(40)	NOT NULL	Default MINEXTENTS, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_MAX_EXTENTS	VARCHAR2(40)	NOT NULL	Default MAXEXTENTS, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_PCT_INCREASE	VARCHAR2(40)	NOT NULL	Default PCTINCREASE, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, for LOCAL index, for ADD TABLE partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELIST GROUPS, for LOCAL index, for ADD TABLE partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING, for LOCAL index, for ADD TABLE PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for LOCAL index, for ADD TABLE PARTITION

ALL_PART_KEY_COLUMNS

This view describes the partitioning key columns for partitioned objects that the current user access.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Partitioned table or index owner
NAME	VARCHAR2(30)		Partitioned table or index name

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(11)		Object type (’TABLE’ or ’INDEX)
COLUMN_NAME	VARCHAR2(30)		Column name
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

ALL_PART_LOBS

This view describes table-level information for partitioned LOBs, including default attributes for LOB data partitions.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of a partitioned table containing LOB(s)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned table containing LOB(s)
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of a LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default CHUNK for a LOB data partition, used for ADD PARTITION
DEF_PCTVERSION	NUMBER	NOT NULL	Default PCTVERSION for a LOB data partition, used for ADD PARTITION
DEF_CACHE	VARCHAR2(3)		Default CACHE for a LOB data partition, used for ADD PARTITION
DEF_IN_ROW	VARCHAR2(3)		Default "IN ROW" for a LOB data partition, used for ADD PARTITION
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default TABLESPACE for a LOB data partition, used for ADD PARTITION
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL for a LOB data partition, used for ADD PARTITION
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT for a LOB data partition, used for ADD PARTITION
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENT for a LOB data partition, used for ADD PARTITION
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS for a LOB data partition, used for ADD PARTITION
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE for a LOB data partition, used for ADD PARTITION

Column	Datatype	NULL	Description
DEF_FREELISTS	VARCHAR2(40)		Default FREELISTS for a LOB data partition, used for ADD PARTITION
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default FREELIST GROUPS for a LOB data partition, used for ADD PARTITION
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition, used for ADD PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default BUFFER POOL for a LOB data partition, used for ADD PARTITION

ALL_PART_TABLES

This view lists the object level partitioning information for partitioned tables the current user access.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of this partitioned table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		In addition to RANGE, may specify HASH
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this table
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method, the default number of subpartitions, if any
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method with Hash subpartitioning, this column will contain a number of columns in the subpartitioning key
DEF_TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Default TABLESPACE, used for add partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, used for add partition
DEF_PCT_USED	NUMBER	NOT NULL	Default PCTUSED, used for add partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, used for add partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, used for add partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL, used for add partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)

Column	Datatype	NULL	Description
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT, used for add partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENTS, used for add partition, 'DEFAULT' if attribute was not specified
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS, used for add partition, 'DEFAULT' if attribute was not specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE, used for add partition, 'DEFAULT' if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, used for add partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELIST GROUPS, used for add partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute, used for add partition
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the given object, used for add partition

ALL_PARTIAL_DROP_TABS

This view describes tables which have partially dropped tables.

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

ALL_POLICIES

This view is a list of all policies on all tables and views that the user has access to.

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
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

Column	Datatype	NULL	Description
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_QUEUES

This view displays information of the queues the user has enqueue or dequeue privilege on. If the user has any of the Advanced Queuing system privileges, like MANAGE ANY QUEUE, ENQUEUE ANY QUEUE or DEQUEUE ANY QUEUE, then all queues in the database will be displayed.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the table the queue data resides in
QID	NUMBER	NOT NULL	Object number of the queue
QUEUE_TYPE	VARCHAR2(15)		Type of the queue
MAX_RETRIES	NUMBER		Maximum number of retries allowed when dequeuing from the queue
RETRY_DELAY	NUMBER		Time interval between retries
ENQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for enqueue
DEQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for dequeue
RETENTION	VARCHAR2(40)		Time interval processed messages retained in the queue
USER_COMMENT	VARCHAR2(50)		User specified comment

ALL_REFRESH

This view lists all the refresh groups that the user can access.

Column	Datatype	NULL	Description
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group

Column	Datatype	NULL	Description
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N; if Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		Y or 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=quick purge option; 2=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

This view lists all the objects in refresh groups, where the user can access the group.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or 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 or 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=quick purge option; 2=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

This view lists the names of the dependent detail or container tables of all the summaries or snapshots 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)		’SNAPSHOT’ or ’SUMMARY’
OLDEST_REFRESH_SCN	NUMBER		The minimum SCN of any summary or snapshot that has TABLE_NAME as a detail table
OLDEST_REFRESH_DATE	DATE		SYSDATE when last refreshed

ALL_REFS

This view describes the REF columns and REF attributes in object type columns accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner

Column	Datatype	NULL	Description
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)		Is the REF value stored with ROWID (YES or NO)?
IS_SCOPED	VARCHAR2(3)		Is the REF column scoped (YES or 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_SNAPSHOTS

This view lists all registered snapshots.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot
SNAPSHOT_SITE	VARCHAR2(128)	NOT NULL	Global name of the snapshot site.
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		Specifies whether the snapshot is updatable. YES if it is, NO if it is not. If set to NO, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Whether the snapshot uses rowids or primary key for fast refresh
SNAPSHOT_ID	NUMBER(38)		Identifier for the snapshot used by the master for fast refresh
VERSION	VARCHAR2(17)		Version of snapshot
QUERY_TXT	LONG		Original query of which this snapshot is an instantiation

ALL_REPCATLOG

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPCOLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPCOLUMN_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPCONFLICT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPDDL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPGENERATED

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPGENOBJECTS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPGROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPGROUPED_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPKEY_COLUMNS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPOBJECT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPPARAMETER_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPPRIORITY

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPPRIORITY_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPPROP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPRESOLUTION

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPRESOL_STATS_CONTROL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPRESOLUTION_METHOD

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPRESOLUTION_STATISTICS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_REPSITES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

ALL_SEQUENCES

This view lists descriptions of sequences accessible to the user.

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_SNAPSHOT_LOGS

This view lists all snapshot logs.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)	NOT NULL	Owner of the log
MASTER	VARCHAR2(30)	NOT NULL	Name of the master table whose changes are logged
LOG_TABLE	VARCHAR2(30)	NOT NULL	Name of the table where the changes to the master table are recorded
LOG_TRIGGER	VARCHAR2(30)		Obsolete with the release of Oracle8*i* Server. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserted rows into the log
ROWIDS	VARCHAR2(3)		If YES, records ROWID information
PRIMARY_KEY	VARCHAR2(3)		If YES, records primary key information
FILTER_COLUMNS	VARCHAR2(3)		If YES, snapshot log records filter columns
CURRENT_SNAPSHOTS	DATE		One date per snapshot; the date the snapshot of the master was last refreshed
SNAPSHOT_ID	NUMBER(38)		Unique identifier of the snapshot

ALL_SNAPSHOT_REFRESH_TIMES

This view lists snapshot refresh times.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

ALL_SNAPSHOTS

This view lists all snapshots accessible to the user.

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

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the view used by users and applications for viewing the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table the snapshot is stored in. This table may have additional columns
MASTER_VIEW	VARCHAR2(30)		View of the master table, owned by the snapshot owner, used for refreshes. This is obsolete in Oracle8*i* and is set to NULL
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table of which this snapshot is a copy
MASTER_LINK	VARCHAR2(128)		Database link name to the master site
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		Specifies whether the snapshot is updatable. YES if it is, NO if it is not. If set to YES, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Values used to drive a refresh of the snapshot (PRIMARY KEY / ROWID / COMPLEX). If PRIMARY KEY, then the snapshot uses primary keys to drive a fast refresh. If ROWID, then it uses RowIDs to drive a fast refresh. If COMPLEX, then fast refreshes are not allowed and the snapshot can only perform complete refreshes
LAST_REFRESH	DATE		Date and time at the master site of the last refresh
ERROR	NUMBER		The number of failed automatic refreshes since last successful refresh
FR_OPERATIONS	VARCHAR2(10)		Status of generated fast refresh operations: (REGENERATE, VALID)
CR_OPERATIONS	VARCHAR2(10)		Status of generated complete refresh operations: (REGENERATE, VALID)
TYPE	VARCHAR2(8)		Type of refresh for all automatic refreshes: COMPLETE, FAST, FORCE
NEXT	VARCHAR2(200)		Date function used to compute next refresh dates
START_WITH	DATE		Date function used to compute next refresh dates
REFRESH_GROUP	NUMBER		All snapshots in a given refresh group get refreshed in the same transaction
UPDATE_TRIG	VARCHAR2(30)		Obsolete. It is NULL for Oracle8*i* snapshots. Formerly, the name of the trigger that fills the UPDATE_LOG

Column	Datatype	NULL	Description
UPDATE_LOG	VARCHAR2(30)		The table that logs changes made to an updatable snapshots
QUERY	LONG		Original query of which this snapshot is an instantiation
MASTER_ROLLBACK_SEG	VARCHAR2(30)		Rollback segment to use at the master site
STATUS	VARCHAR2(7)		The status of the contents of the snapshot
REFRESH_MODE	VARCHAR2(8)		This indicates how and when the snapshot will be refreshed
PREBUILT	VARCHAR2(3)		If YES, this snapshot uses a prebuilt table as the base table

ALL_SOURCE

This view lists the text source of all stored objects accessible to the 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 object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY, 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_SUBPART_COL_STATISTICS

This view lists column statistics and histogram information for table subpartitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
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

Column	Datatype	NULL	Description
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying subpartitions?
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

This view lists the actual histogram data (end-points per histogram) for histograms on table subpartitions.

SUB

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

ALL_SUBPART_KEY_COLUMNS

This view lists subpartitioning key columns for tables (and Local indexes on tables) partitioned using the Composite Range/Hash method.

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	Datatype	NULL	Description
COLUMN_NAME	VARCHAR2(30)		The column name
COLUMN_POSITION	NUMBER		The position of the column within the subpartitioning key

ALL_SUMDELTA

This view 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_SUMMARIES

This view represents a summary object.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym
SUMMARY_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym
CONTAINER_OWNER	VARCHAR2(30)	NOT NULL	Owner of the container object
CONTAINER_NAME	VARCHAR2(30)		Name of the table or snapshot that the summary data is contained in
CONTAINER_TYPE	VARCHAR2(9)		’TABLE’ or ’SNAPSHOT’
LAST_REFRESH_SCN	NUMBER		System Change Number
LAST_REFRESH_DATE	DATE		SYSDATE of last refresh
REFRESH_METHOD	VARCHAR2(11)		’ANY’, ’INCREMENTAL’, ’FULL’, or ’NONE’
FULLREFRESHTIM	NUMBER		The time that it took to fully refresh the summary
INCREFRESHTIM	NUMBER		The time that it took to incrementally refresh the summary

Column	Datatype	NULL	Description
CONTAINS_VIEWS	VARCHAR2(1)		This summary contains views in the FROM clause
UNUSABLE	VARCHAR2(1)		This summary is unusable, the build was deferred
RESTRICTED_SYNTAX	VARCHAR2(1)		Y if this summary uses restricted syntax, else 'N'
INC_REFRESHABLE	VARCHAR2(1)		Y if this summary can be incrementally refreshed, else 'N
KNOWN_STALE	VARCHAR2(1)		Y if the data contained in this summary is known to be inconsistent with the detail data, and 'N' otherwise
QUERY_LEN	NUMBER	NOT NULL	Length of the query field
QUERY	LONG	NOT NULL	SELECT expression of the summary definition

ALL_SYNONYMS

This view lists all synonyms accessible to the user.

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

This view contains column statistics and histogram information which is in the USER_TAB_COLUMNS view. For more information, see "USER_TAB_COLUMNS" on page 2-249.

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

Column	Datatype	NULL	Description
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
LAST_ANALYZED	DATE		Date of the most recent time this column was analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
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

This view lists the columns of all tables, views, and clusters accessible to the user. To gather statistics for this view, use the SQL command ANALYZE.

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
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

Column	Datatype	NULL	Description
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		The number of nulls in the column
NUM_BUCKETS	NUMBER		The number of buckets in histogram for the column
LAST_ANALYZED	DATE		The date of the most recent time this column was 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)		Are the statistics calculated without merging underlying partitions?
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_COMMENTS

This view lists comments on tables and views accessible to the user.

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

This view lists histograms on tables and views accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

ALL_TAB_PARTITIONS

This view describes, for each table partition, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE that the current user can access.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COMPOSITE	VARCHAR2(3)		'YES' if the partition belongs to a Local index on a table partitioned using Composite method; 'NO' otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
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

Column	Datatype	NULL	Description
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 of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for this partition
GLOBAL_STATS	VARCHAR2(3)		Global statistics
USER_STATS	VARCHAR2(3)		User statistics

ALL_TAB_PRIVS

This view lists the grants on objects for which the user or PUBLIC is the grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access is granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_PRIVS_MADE

This view lists the user’s grants and grants on the user’s objects.

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_PRIVS_RECD

This view lists grants on objects for which the user or PUBLIC is the grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_SUBPARTITIONS

This view describes, for each table subpartition, its name, name of the table and partition to which it belongs, and its storage attributes. Note that statistics will not be 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

Column	Datatype	NULL	Description
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of 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 when last analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

ALL_TABLES

This view contains descriptions of relational tables accessible to the user. To gather statistics for this view, use the SQL command ANALYZE.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned 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
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
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

Column	Datatype	NULL	Description
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 table is to be cached in the buffer cache
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 of the most recent time this table was 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 or IOT_OVERFLOW. If this is not an index organized table, then IOT_TYPE is NULL
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2(1)		Is the object created as part of icreate for domain indexes?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object; NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL

Column	Datatype	NULL	Description
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

ALL_TRIGGERS

This view lists trigger information for triggers owned by the user, triggers on tables owned by the user, or all triggers if the user has the CREATE ANY TRIGGER privilege.

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(75)		Events that fire the trigger: INSERT, UPDATE, DELETE, STARTUP, SHUTDOWN, ERROR, LOGON, LOGOFF, CREATE, ALTER, DROP
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
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 or 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_TRIGGER_COLS

This view displays the usage of columns in triggers owned by user, on tables owned by user, or on all triggers if the user has the CREATE ANY TRIGGER privilege.

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_TYPE_ATTRS

This view displays the attributes of types accessible to the user.

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
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)		The name of the character set: 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 ID number)

ALL_TYPE_METHODS

This view is a description of methods of types accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	Method number for distinguishing overloaded method (not to be used as ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method

ALL_TYPES

This view displays the types accessible to the user.

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 in the type
METHODS	NUMBER		Number of methods in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type

ALL_UNUSED_COL_TABS

This view contains a description of all tables containing unused columns.

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

This view contains a description of all columns that are updatable in a join view.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

ALL_USERS

This view contains information about all users of the database.

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

This view contains information about 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
PROPERTY	VARCHAR2(6)		Property of the object - column or index
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

ALL_VARRAYS

This view lists the text of views accessible to the user.

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

ALL_VIEWS

This view lists the text of views accessible to the user.

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

AUDIT_ACTIONS

This view contains descriptions for audit trail action type codes.

Column	Datatype	NULL	Description
ACTION	NUMBER	NOT NULL	Numeric audit trail action type code
NAME	VARCHAR2(27)	NOT NULL	Name of the type of audit trail action

CATALOG

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

CAT

This is a synonym for USER_CATALOG. For more information, see "USER_CATALOG" on page 2-196.

CHAINED_ROWS

Stores the output for the ANALYZE command with the CHAINED ROWS option.

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 command was issued

CLU

This is a synonym for USER_CLUSTERS. For more information, see "USER_CLUSTERS" on page 2-196.

CODE_PIECES

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" on page 2-110 and "USER_OBJECT_SIZE" on page 2-224.

CODE_SIZE

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" on page 2-110 and "USER_OBJECT_SIZE" on page 2-224.

COL

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

COLS

This is a synonym for USER_TAB_COLUMNS. For more information, see "USER_TAB_COLUMNS" on page 2-249.

COLUMN_PRIVILEGES

This view lists grants on columns 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. Use of this view is not recommended.

Column	Description
GRANTEE	Name of the user to whom access was granted.
OWNER	Username of the object's owner.
TABLE_NAME	Name of the object.
COLUMN_NAME	Name of the column.
GRANTOR	Name of the user who performed the grant.
INSERT_PRIV	Permission to insert into the column.
UPDATE_PRIV	Permission to update the column.
REFERENCES_PRIV	Permission to reference the column.
CREATED	Timestamp for the grant.

Column	Description
GRANTEE	Name of the user to whom access was granted
OWNER	Username of the object's owner
TABLE_NAME	Name of the object
COLUMN_NAME	Name of the column
GRANTOR	Name of the user who performed the grant
INSERT_PRIV	Permission to insert into the column
UPDATE_PRIV	Permission to update the column
REFERENCES_PRIV	Permission to reference the column
CREATED	Timestamp for the grant

DBA_2PC_NEIGHBORS

This view contains information about 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: client database name; OUT: outgoing database link

Column	Datatype	NULL	Description
DBUSER_OWNER	VARCHAR2(30)		IN: name of local user; OUT: 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

This view contains information about 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 = 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 (tx or system recovery)
FORCE_TIME	DATE		Time of manual force decision (null if not forced locally)
RETRY_TIME	DATE	NOT NULL	Time automatic recovery (RECO) last tried to recover the transaction
OS_USER	VARCHAR2(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

This view displays descriptions of all tables (object tables and relational tables) in the database.

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)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		The average freespace of all blocks on a freelist

Column	Datatype	NULL	Description
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)		Type of the table if the table is a typed table
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2(1)		Is the index object created as part of icreate for domain indexes?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		The default buffer pool to be used for table blocks
ROW_MOVEMENT	VARCHAR2(8)		The movement of the row
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

DBA_ANALYZE_OBJECTS

This view lists analyze objects.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner
OBJECT_NAME	VARCHAR2(30)	NOT NULL	The object name
OBJECT_TYPE	VARCHAR2(7)		The object type

DBA_ASSOCIATIONS

This view lists user-defined statistics information.

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	Object name 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)		Schema type of the object - column, type, package or function
STATSTYPE_SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of Statistics type which contains the cost, selectivity or stats funcs
DEF_SELECTIVITY	NUMBER		Default Selectivity if any of the object
DEF_CPU_COST	NUMBER		Default CPU cost if any of the object
DEF_IO_COST	NUMBER		Default I/O cost if any of the object
DEF_NET_COST	NUMBER		Default Networking cost if any of the object

DBA_AUDIT_EXISTS

This view lists audit trail entries produced by AUDIT NOT EXISTS and AUDIT EXISTS.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(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
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: ▪ zero: the action succeeded ▪ 2004: security violation

DBA_AUDIT_OBJECT

This view contains audit trail records for all objects in the system.

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

Column	Datatype	NULL	Description
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_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: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_AUDIT_SESSION

This view lists all audit trail records concerning CONNECT and DISCONNECT.

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: ■ zero: the action succeeded ■ 2004: security violation

DBA_AUDIT_STATEMENT

This view lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements.

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

Column	Datatype	NULL	Description
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
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_AUDIT_TRAIL

This view lists all audit trail entries.

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
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

Column	Datatype	NULL	Description
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: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_BLOCKERS

This view lists all sessions that have someone waiting on a lock they hold that are not themselves waiting on a lock.

Column	Datatype	NULL	Description
HOLDING_SESSION	NUMBER		Session holding a lock

DBA_CATALOG

This view lists all database tables, views, synonyms, and sequences.

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

Column	Datatype	NULL	Description
TABLE_TYPE	VARCHAR2(11)		Type of the object

DBA_CLU_COLUMNS

This view lists mappings of table columns to cluster columns.

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_CLUSTERS

This view contains description of all clusters in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing 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
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment

Column	Datatype	NULL	Description
AVG_BLOCKS_PER_KEY	NUMBER		Average number of blocks containing rows with a given cluster key
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)		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 table is to be cached in the buffer cache
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
SINGLE_TABLE	VARCHAR2(5)		Y if the cluster is single table; N if not

DBA_CLUSTER_HASH_EXPRESSIONS

This view list hash functions for all clusters.

This view list all clusters accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of owner of cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of cluster
HASH_EXPRESSION	LONG		Text of hash function of cluster

DBA_COL_COMMENTS

This view lists comments on columns of all tables and views.

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
COMMENTS	VARCHAR2(4000)		Comment on the object

DBA_COL_PRIVS

This view lists all grants on columns in the database.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Username 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	Column privilege
GRANTABLE	VARCHAR2(3)		Privilege is Grantable

DBA_COLL_TYPES

This view displays all named collection types in the database such as VARRAYs, nested tables, object tables, and so on.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
COLL_TYPE	VARCHAR2(30)	NOT NULL	Collection type
UPPER_BOUND	NUMBER		Maximum size of the VARRAY type
ELEM_TYPE_MOD	VARCHAR2(7)		Type modifier of the element
ELEM_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the element
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the type of the element
LENGTH	NUMBER		Length of the CHAR element or maximum length of the VARCHAR or VARCHAR2 element
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL element or binary precision of the FLOAT element
SCALE	NUMBER		Scale of the NUMBER or DECIMAL element
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS NCHAR_CS
ELEM_STORAGE	VARCHAR2(7)		Storage optimization specification for VARRAY of numeric elements
NULLS_STORED	VARCHAR2(3)		Is null information stored with each VARRAY element?

DBA_CONSTRAINTS

This view contains constraint definitions on all tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with 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 table with constraint definition
SEARCH_CONDITION	LONG		Text of search condition for table check
R_OWNER	VARCHAR2(30)		Owner of table used in referential constraint
R_CONSTRAINT_NAME	VARCHAR2(30)		Name of unique constraint definition for referenced table
DELETE_RULE	VARCHAR2(9)		Delete rule for a referential constraint: CASCADE / NO ACTION
STATUS	VARCHAR2(8)		Enforcement status of constraint: ENABLED or DISABLED
DEFERRABLE	VARCHAR2(14)		Indicates whether the constraint is deferrable
DEFERRED	VARCHAR2(9)		Indicates whether the constraint was initially deferred
VALIDATED	VARCHAR2(13)		Indicates whether all data obeys the constraint: VALIDATED, NOT VALIDATED
GENERATED	VARCHAR2(14)		Indicates whether the name system is generated
BAD	VARCHAR2(3)		Creating this constraint should give ORA-02436. Rewrite it before 2000 AD.
RELY	VARCHAR2(4)		If set, this flag will be used in the optimizer
LAST_CHANGE	DATE		Indicates when the constraint was last enabled or disabled

DBA_CONS_COLUMNS

This view contains information about accessible columns in constraint definitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name associated with table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name associated with column or attribute of the object type column specified in the constraint definition
POSITION	NUMBER		Original position of column or attribute in definition

Note: If you create a constraint on a user-defined REF column, the system creates the constraint on the exploded columns that make up the REF column and hence column names of the constraint will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

DBA_CONTEXT

This view lists all context namespaces information.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	The name of the context namespace
SCHEMA	VARCHAR2(30)	NOT NULL	The schema name of the designated package which can set attributes using this namespace
PACKAGE	VARCHAR2(30)	NOT NULL	The package name of the designated package which can set attributes using this namespace

DBA_DATA_FILES

This view contains information about 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

Column	Datatype	NULL	Description
BLOCKS	NUMBER	NOT NULL	Size of the file in Oracle blocks
STATUS	VARCHAR2(9)		File status: AVAILABLE or INVALID (INVALID means that the file number is not in use, for example, a file in a tablespace that was dropped)
RELATIVE_FNO	NUMBER		Relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator
MAXBYTES	NUMBER		Maximum file size in bytes
MAXBLOCKS	NUMBER		Maximum file size in blocks
INCREMENT_BY	NUMBER		Autoextension increment
USER_BYTES	NUMBER		Corresponding number of bytes
USER_BLOCKS	NUMBER		Number of blocks which can be used by the data

DBA_DB_LINKS

This view lists all database links in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the database link
DB_LINK	VARCHAR2(128)	NOT NULL	Name of the database link
USERNAME	VARCHAR2(3)		Name of user to log in as
HOST	VARCHAR2(2000)		Connect string
CREATED	DATE	NOT NULL	Creation time of the database link

DBA_DDL_LOCKS

This view 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

Column	Datatype	NULL	Description
MODE_HELD	VARCHAR2(9)		Lock mode: NONE, NULL, SHARE, EXCLUSIVE
MODE_REQUESTED	VARCHAR2(9)		Lock request type: NONE, NULL, SHARE, EXCLUSIVE

DBA_DEPENDENCIES

This view lists dependencies to and from objects. Dependencies on views created without any database links are also available.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(3)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY.
REFERENCED_OWNER	VARCHAR2(30)		Owner of referenced object (remote owner if remote object)
REFERENCED_NAME	VARCHAR2(64)		Type of parent object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
REFERENCED_TYPE	VARCHAR2(12)		Type of referenced object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY.
REFERENCED_LINK_NAME	VARCHAR2(128)		Name of dblink if this is a remote object
DEPENDENCY_TYPE	VARCHAR2(4)		Two values: REF when the dependency is a REF dependency; HARD otherwise

DBA_DIMENSIONS

This view represents dimension objects.

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)		’Y’ if this dimension is in an invalid state, else ’N’
REVISION	NUMBER		Dimension revision level

DBA_DIM_ATTRIBUTES

This view represents the relationship between a dimension level and a functionally dependent column. The table that the level columns are in must match the table of the dependent column.

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

DBA_DIM_CHILD_OF

This view represents a 1:n hierarchical relationship between a pair of levels in a dimension.

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

DBA_DIM_JOIN_KEY

This view represents a join between two dimension tables. The join is always specified between a parent dimension level column and a child column.

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	Unique within a dimension

Column	Datatype	NULL	Description
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 key column
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column

DBA_DIM_HIERARCHIES

This view represents a dimension hierarchy.

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

DBA_DIM_LEVELS

This view represents a dimension level. All columns of a dimension level must come from the same relation.

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

DBA_DIM_LEVEL_KEY

This view represents a column of a dimension level. The position of a column within a level is specified by KEY_POSITION.

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

Column	Datatype	NULL	Description
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

DBA_DIRECTORIES

This view provides information on all directory objects 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

DBA_DML_LOCKS

This view 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)		Lock mode: see Table 2–1, "Lock Modes for the DBA_DML_LOCKS View"
MODE_REQUESTED	VARCHAR2(13)		Lock request type: see Table 2–1, "Lock Modes for the DBA_DML_LOCKS View"
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Blocking others

Table 2–1 describes DML lock mode values that are valid for the MODE_HELD column.

Table 2–1 Lock Modes for the DBA_DML_LOCKS View

Lock Mode	Description
ROWS-S (SS)	Row share
ROW-X (SX)	Row exclusive
SHARE (S)	Share
S/ROW-X (SSX)	Exclusive
NONE	MODE_HELD: Lock requested, not yet obtained MODE_REQUESTED: Lock identifier obtained, lock not held or requested

DBA_ERRORS

This view lists current errors on all stored objects in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: VIEW, PROCEDURE, FUNCTION, PACKAGE, TYPE, TYPE BODY, PACKAGE BODY, or TRIGGER
SEQUENCE	NUMBER	NOT NULL	Sequence number used for ordering purposes
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

DBA_EXP_FILES

This view contains a description of 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
TIMESTAMP	DATE	NOT NULL	Timestamp of the export session

DBA_EXP_OBJECTS

This view lists 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

This view contains the version number of the last export session.

Column	Datatypes	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the last export session

DBA_EXTENTS

This view lists the extents comprising all segments in the database.

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
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

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)

DBA_FREE_SPACE

This view lists the free extents in all tablespaces.

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
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

This view 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_INDEXES

This view contains descriptions for all indexes in the database. To gather statistics for this view, use the SQL command ANALYZE. This view supports parallel partitioned index scans.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username of the owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(12)		Type of index
TABLE_OWNER	VARCHAR2(30)		Owner of the indexed object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
TABLE_TYPE	VARCHAR2(11)	NOT NULL	Type of the indexed object
UNIQUENESS	VARCHAR2(9)		Uniqueness status of the index: UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(11)		Enabled or disabled
PREFIX_LENGTH	NUMBER		The number of columns in the prefix of the key used for compression
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 initial extent
NEXT_EXTENT	NUMBER		Size of secondary extents
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		User column-id for last column to be included in index organized table top index
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 attribute

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		The number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		The number of distinct keys in the index
AVG_LEAF_BLOCKS _PER_KEY	NUMBER		The average number of leaf blocks per key
AVG_DATA_BLOCKS _PER_KEY	NUMBER		The average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		A measurement of the amount of (dis)order of the table this index is for
STATUS	VARCHAR2(8)		State of the index: DIRECT LOAD, VALID or INPROGS (a DDL operation on the domain index is in progress).
DOMIDX_STATUS	VARCHAR		Reflects the status of the domain index. A NULL value means that the specified index is not a domain index. A Value of VALID means that the index is a domain index and the index does not have any errors. If the value of this column is IDXTYP_INVLD it means that the indextype corresponding to this domain index is invalid.
DOMIDX_OPSTATUS	VARCHAR		Reflects the status of an operation that was performed on the domain index. A value of NULL indicates that the specified index is not a domain index. A value of VALID specifies that the index does not have any errors. A value of FAILED indicates that the operation that was performed on the domain index failed with an error.
FUNCIDX_STATUS	VARCHAR		A value of NULL indicates a non function-based index. ENABLED indicates the function-based index is enabled. DISABLED indicates the function-based index is disabled.
NUM_ROWS	NUMBER		Number of rows in this index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze this index
LAST_ANALYZED	DATE		Timestamp for when this index was last analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index. NULL if PARTITIONED=NO.
INSTANCES	VARCHAR2(40)		Number of instances across which the indexes are to be scanned. NULL if PARTITIONED=NO.
PARTITIONED	VARCHAR2(3)		Indicates whether this index is partitioned. Set to YES if it is partitioned

Column	Datatype	NULL	Description
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this index system generated?
SECONDARY	VARCHAR2(1)		Is the index object created as part of icreate for domain indexes?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		The duration
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

DBA_IND_COLUMNS

This view contains descriptions of the columns comprising the indexes on all tables and clusters.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column or attribute
DESCEND	VARCHAR2(4)		Y/N, Y if this column is sorted in descending order

Note: If you create an index on a user-defined REF column, the system creates the index on the exploded columns that make up the REF column and hence column names of the index will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

DBA_IND_EXPRESSIONS

This view lists functional index expressions on all tables and clusters.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Functional index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

DBA_IND_PARTITIONS

This view describes, for each index partition, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
COMPOSITE	VARCHAR2(3)		’YES’ if the partition belongs to a Local index on a table partitioned using Composite method; ’NO’ otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the index
STATUS	VARCHAR2(8)		Indicates 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

Column	Datatype	NULL	Description
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for
NUM_ROWS	NUMBER		Number of rows in this index partition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The buffer pool for this partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

DBA_IND_SUBPARTITIONS

This view describes, for each index subpartition that the current user owns, the partition level partitioning information, the storage parameters for the subpartition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	The owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index 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
STATUS	VARCHAR2(8)		Indicates whether index partition is usable or not
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key
AVG_DATA_BLOCKS _PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for

Column	Datatype	NULL	Description
NUM_ROWS	NUMBER		Number of rows in this index partition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The buffer pool for the partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

DBA_INDEXTYPES

This view lists all the indextypes.

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
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of indextype implementation
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype

DBA_INDEXTYPE_OPERATORS

This view lists all the operators supported by indextypes.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
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

DBA_JOBS

This view lists all jobs in the database.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change it.
LOG_USER	VARCHAR2(30)	NOT NULL	USER who was logged in 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 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 (usually NULL if not executing)
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE.
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2(8))		Same as NEXT_DATE. The job becomes due for execution at this time.
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job, in seconds
BROKEN	VARCHAR2(1)		If Y, no 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		How many times has this job 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)		ALTER SESSION parameters describing the NLS environment of the job
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0

DBA_JOBS_RUNNING

This view 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 job started executing.
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0

DBA_LIBRARIES

This view lists all the libraries in the database.

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)		Is the library dynamically loadable? (YES or NO)
STATUS	VARCHAR2(7)		Status of the library

DBA_LOBS

This view displays the LOBs contained in all tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table containing the LOB
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table containing the LOB
COLUMN_NAME	VARCHAR2(30)		Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB segment

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB index
CHUNK	NUMBER		Size of the LOB chunk as a unit of allocation / manipulation in bytes
PCTVERSION	NUMBER	NOT NULL	Maximum percentage of the LOB space used for versioning
CACHE	VARCHAR2(3)		Indicates whether the LOB is accessed through the buffer cache
LOGGING	VARCHAR2(3)		Indicates whether the changes to the LOB are logged
IN_ROW	VARCHAR2(3)		Are some of the LOBs stored with the base row?

DBA_LOB_PARTITIONS

This view displays the LOBs contained in tables accessible to the user.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Table owner
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a partitioned LOB item
PARTITION_NAME	VARCHAR2(30)		Name of a table partition
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the lOB item
COMPOSITE	VARCHAR2(3)		Is it a Composite partition (one of YES, NO)?
CHUNK	NUMBER		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size of the initial extent in bytes for a non-composite LOB data partition or size of the initial extent in blocks for a composite LOB data partition (to ascertain the number of bytes, query DBA_LOB_SUBPARTITIONS)

Column	Datatype	NULL	Description
NEXT_EXTENT	VARCHAR2(40)		Size of secondary extents in bytes for a LOB data partition or size of the secondary extents in blocks for a composite LOB data partition (to ascertain the number of bytes, query DBA_LOB_SUBPARTITIONS)
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for a LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of a LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

DBA_LOB_SUBPARTITIONS

This view displays partition-level attributes of LOB data subpartitions.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Table owner
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a 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 a table subpartition to which this LOB subpartition corresponds
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		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes for a LOB data partition
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes for a LOB data partition
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	NUMBER		Percentage increase in extent size for a LOB data partition
FREELISTS	NUMBER		Number of process freelists allocated in the segment of a LOB data partition
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

DBA_LOCKS

This view lists all locks or latches held in the database, and all outstanding requests for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(26)		Lock type
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)		Blocking others

DBA_LOCK_INTERNAL

This view has 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
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_METHOD_PARAMS

This view is a description of method parameters of types in the database.

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)
PARAM_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number or position
PARAM_MODE	VARCHAR2(6)		Mode of the parameter
PARAM_TYPE_MOD	VARCHAR2(7)		Type modifier of the parameter
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)		The name of the character set: CHAR_CS NCHAR_CS

DBA_METHOD_RESULTS

This view is a description of method results of all types in the database.

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

Column	Datatype	NULL	Description
METHOD_NO	NUMBER	NOT NULL	The method number
RESULT_TYPE_MOD	VARCHAR2(7)		Type modifier of the result
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)		The name of the character set: CHAR_CS, NCHAR_CS

DBA_MVIEW_AGGREGATES

This view represents the grouping functions (aggregated measures) that appear in the SELECT list of an aggregated materialized view.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view index
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Position of this measure within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Set to Y if DISTINCT aggregation
MEASURE	LONG		Contains the SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*)

DBA_MVIEW_ANALYSIS

This view represents the materialized views that potentially support query rewriting and that have additional information that is available for analysis by applications. This view excludes any materialized view that references a remote table or that includes a reference to a non-static value such as SYSDATE or USER. It does show all materialized views upon which Oracle would allow ENABLE QUERY REWRITE, not just those that are actually enabled for 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
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view table object

Column	Datatype	Null	Description
CONTAINER_NAME	VARCHAR2(30)		Name of the snapshot that the materialized view data is contained in
LAST_REFRESH_SCN	NUMBER		The System Change Number
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		FORCE, FAST, COMPLETE, or NONE
SUMMARY	VARCHAR2(1)		Y if this materialized view includes a GROUP BY clause or aggregation, else N
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
INCREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
CONTAINS_VIEWS	VARCHAR2(1)		Y if this materialized view contains a view in its definition, else N
UNUSABLE	VARCHAR2(1)		Y if this materialized view is in an unusable state (inconsistent data), else N. A materialized view can be in an unusable state if a system failure occurs during a full refresh
RESTRICTED_SYNTAX	VARCHAR2(1)		Y if this materialized view uses restricted materialized view syntax, else N
INC_REFRESHABLE	VARCHAR2(1)		Y if this materialized view can be incrementally refreshed, else N
KNOWN_STALE	VARCHAR2(1)		Y if the data contained in the materialized view is known to be inconsistent with the detail data, else N
INVALID	VARCHAR2(1)		Y if this materialized view is in an invalid state (inconsistent metadata), else N
QUERY_LEN	NUMBER	NOT NULL	The length of the query field
QUERY	LONG	NOT NULL	SELECT expression of the materialized view definition
REVISION	NUMBER		The materialized view metadata revision level

DBA_MVIEW_DETAIL_RELATIONS

This view 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. Inline views in the materialized view definition are not represented in this table.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name

Column	Datatype	Null	Description
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, MATERIALIZED VIEW, or SNAPSHOT
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

DBA_MVIEW_JOINS

This view represents a join between two columns in the WHERE clause of a materialized view.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or TABLE
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

DBA_MVIEW_KEYS

This view 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. Inline views in the materialized view definition are not represented in this table.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view

Column	Datatype	Null	Description
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or TABLE
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

DBA_NESTED_TABLES

This view displays descriptions of the nested tables contained in all tables.

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)		Indication of default or user-specified storage for the varray
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column locator or value

DBA_OBJECT_SIZE

This view lists the sizes, in bytes, of various PL/SQL objects.

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

Column	Datatype	NULL	Description
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

This view displays descriptions of all object tables in the database.

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)	NOT NULL	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)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table

Column	Datatype	NULL	Description
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)	NOT NULL	Type of the table if the table is a typed table
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		The 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)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

DBA_OBJECTS

This view lists all objects in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the owner of the object
OBJECT_NAME	VARCHAR2(128)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the sub-object (for example, partition)
OBJECT_ID	NUMBER		Object number of the object
DATA_OBJECT_ID	NUMBER		Object number of the segment which contains the object
OBJECT_TYPE	VARCHAR2(15)		Type of the object (eg, TABLE, INDEX)
CREATED	DATE		Timestamp for the creation of the object
LAST_DDL_TIME	DATE		Timestamp for the last DDL change (including GRANT and REVOKE) to the object
TIMESTAMP	VARCHAR2(20)		Timestamp for the specification of the object
STATUS	VARCHAR2(7)		Status of the object
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this object system generated?
SECONDARY	VARCHAR2(1)		Is this a secondary object created as part of icreate for domain indexes

DBA_OBJ_AUDIT_OPTS

This view lists auditing options for all objects owned by a user.

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

Column	Datatype	NULL	Description
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 EXE WHENEVER SUCCESSFUL / UNSUCCESSFUL
CRE	VARCHAR2(3)		Auditing CRE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing REA WHENEVER SUCCESSFUL / UNSUCCESSFUL
WRI	VARCHAR2(3)		Auditing WRI WHENEVER SUCCESSFUL / UNSUCCESSFUL

DBA_OPANCILLARY

This view lists ancillary information for operator bindings.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of primary operator

DBA_OPARGUMENTS

This view lists argument information for operator bindings.

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

Column	Datatype	NULL	Description
POSITION	NUMBER	NOT NULL	Position of the operator argument
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

DBA_OPBINDINGS

This view lists operator bindings.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		Name of the schema of the return type - not null only for ADTs
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		Schema of the implementation type of the indextype
IMPLEMENTATION_TYPE	VARCHAR2(30)		Implementation type of the indextype

DBA_OPERATORS

This view lists operators.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

DBA_OUTLINE_HINTS

This view lists the set of hints which make up the outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		The name of the user who created the outline
NODE	NUMBER		I.D. of the query or subquery to which the hint applies. The top-level query is labelled 1. Subqueries are assigned sequentially numbered labels, starting with 2
JOIN_POS	NUMBER		Position of the table in the join order. The JOIN_POS column is 0 for all hints except the access method hints. The access method hints identify a table to which the hint and the join position apply
HINT	VARCHAR2(512)		Text of the hint

DBA_OUTLINES

This view lists information about outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		The 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)		The name of the user who created the outline
CATEGORY	VARCHAR2(30)		A user-defined name used to group outlines into collections
USED	VARCHAR2(9)		Flag indicating whether the outline has ever been used
TIMESTAMP	DATE		Timestamp at which the outline was created
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

DBA_PART_COL_STATISTICS

This view contains column statistics and histogram information for all table partitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2(30)		Table partition name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name

Column	Datatype	NULL	Description
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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

DBA_PART_HISTOGRAMS

This view contains the histogram data (end-points per histogram) for histograms on all table partitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner name
TABLE_NAME	VARCHAR2(30)		Table name
PARTITION_NAME	VARCHAR2(30)		Table partition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

DBA_PART_INDEXES

This view lists the object level partitioning information for all partitioned indexes.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of this partitioned index

Column	Datatype	NULL	Description
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)		In addition to RANGE, may specify HASH
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this index
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	Number of partitions in this index
PARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
LOCALITY	VARCHAR2(6)		Indicates whether this partitioned index is LOCAL or GLOBAL
ALIGNMENT	VARCHAR2(12)		Indicates whether this partitioned index is PREFIXED or NON-PREFIXED
DEF_TABLESPACE _NAME	VARCHAR2(30)		Default TABLESPACE, used for LOCAL index, for ADD / SPLIT TABLE PARTITION
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, used for LOCAL index, for ADD TABLE PARTITION
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, used for LOCAL index, for ADD TABLE PARTITION
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, used for LOCAL index, for ADD TABLE PARTITION
DEF_INITIAL_EXTENT	NUMBER		Default INITIAL, used for LOCAL index, for ADD TABLE PARTITION, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_NEXT_EXTENT	NUMBER		Default NEXT, used for LOCAL index, for ADD TABLE PARTITION, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	NUMBER		Default MINEXTENTS, used for LOCAL index, for ADD TABLE PARTITION, ’DEFAULT’ if attribute was not specified
DEF_MAX_EXTENTS	NUMBER		Default MAXEXTENTS, used for LOCAL index, for ADD TABLE PARTITION, ’DEFAULT’ if attribute was not specified
DEF_PCT_INCREASE	NUMBER		Default PCTINCREASE, used for LOCAL index, for ADD TABLE PARTITION, ’DEFAULT’ if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, used for LOCAL index, for ADD TABLE PARTITION

Column	Datatype	NULL	Description
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELISTS, used for LOCAL index, for ADD TABLE PARTITION
DEF_LOGGING	VARCHAR2(7)		Default LOGGING, for LOCAL index, for ADD TABLE PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the index, for ADD TABLE PARTITION

DBA_PART_KEY_COLUMNS

This view describes the partitioning key columns for all partitioned objects.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Partitioned table or index owner
NAME	VARCHAR2(30)		Partitioned table or index name
OBJECT_TYPE	VARCHAR2(11)		The object type ('TABLE' or 'INDEX')
COLUMN_NAME	VARCHAR2(30)		Column name
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

DBA_PART_LOBS

This view describes table-level information for partitioned LOBs, including default attributes for LOB data partitions.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of a partitioned table containing LOB(s)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned table containing LOB(s)
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of a LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default CHUNK for a LOB data partition, used for ADD PARTITION
DEF_PCTVERSION	NUMBER	NOT NULL	Default PCTVERSION for a LOB data partition, used for ADD PARTITION
DEF_CACHE	VARCHAR2(3)		Default CACHE for a LOB data partition, used for ADD PARTITION

Column	Datatype	NULL	Description
DEF_IN_ROW	VARCHAR2(3)		Default "IN ROW" for a LOB data partition, used for ADD PARTITION
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default TABLESPACE for a LOB data partition, used for ADD PARTITION
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL for a LOB data partition, used for ADD PARTITION
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT for a LOB data partition, used for ADD PARTITION
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENT for a LOB data partition, used for ADD PARTITION
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS for a LOB data partition, used for ADD PARTITION
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE for a LOB data partition, used for ADD PARTITION
DEF_FREELISTS	VARCHAR2(40)		Default FREELISTS for a LOB data partition, used for ADD PARTITION
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default FREELIST GROUPS for a LOB data partition, used for ADD PARTITION
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition, used for ADD PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default BUFFER POOL for a LOB data partition, used for ADD PARTITION

DBA_PART_TABLES

This view lists the object level partitioning information for all the partitioned tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of this partitioned table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		In addition to RANGE, may specify HASH
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this table
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method, the default number of subpartitions, if any
PARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key

Column	Datatype	NULL	Description
SUBPARTITIONING_KEY_COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method with Hash subpartitioning, this column will contain a number of columns in the subpartitioning key
DEF_TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Default TABLESPACE, used for ADD partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, used for ADD partition
DEF_PCT_USED	NUMBER	NOT NULL	Default PCTUSED, used for ADD partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, used for ADD partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, used for ADD partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL, used for ADD partition, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT, used for ADD partition, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENTS, used for ADD partition, ’DEFAULT’ if attribute was not specified
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS, used for ADD partition, ’DEFAULT’ if attribute was not specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE, used for ADD partition, ’DEFAULT’ if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, used for ADD partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELIST GROUPS, used for ADD partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute, used for ADD partition
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the given object, used for ADD partition

DBA_PARTIAL_DROP_TABS

This view describes tables which have partially dropped tables.

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

DBA_PENDING_TRANSACTIONS

This view gives information about unresolved transactions (either due to failure or if the co-ordinator has not sent a commit / rollback).

Column	Datatype	NULL	Description
FORMATID	NUMBER		The format identifier of the transaction identifier
GLOBALID	RAW(64)		The global part (gtrid) of the transaction identifier
BRANCHID	RAW(64)		The branch qualifier (bqual) of the transaction identifier

DBA_POLICIES

This view lists 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
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?

DBA_PRIV_AUDIT_OPTS

This view 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

Column	Datatype	NULL	Description
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_PROFILES

This view displays all profiles and their limits.

Column	Datatype	NULL	Description
PROFILE	VARCHAR2(30)	NOT NULL	Profile name
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Resource name
RESOURCE_TYPE	VARCHAR2(8)		Indicates whether the resource profile is a KERNEL or a PASSWORD parameter
LIMIT	VARCHAR2(40)		Limit placed on this resource for this profile

DBA_QUEUE_SCHEDULES

This view describes the current schedules for propagating messages.

Column	Datatype	NULL	Description
SCHEMA	VARCHAR2(30)	NOT NULL	The schema
QNAME	VARCHAR2(30)	NOT NULL	Source queue name
DESTINATION	VARCHAR2(128)	NOT NULL	Destination name, currently limited to be a DBLINK name
START_DATE	DATE		Date to start propagation in the default date format
START_TIME	VARCHAR2(8)		Time of day at which to start propagation in HH:MI:SS format
PROPAGATION _WINDOW	NUMBER		Duration in seconds for the propagation window
NEXT_TIME	VARCHAR2(200)		Function to compute the start of the next propagation window
LATENCY	NUMBER		Maximum wait time to propagate a message during the propagation window
SCHEDULE_DISABLED	VARCHAR2(1)		N if enabled; Y if disabled and schedule will not be executed
PROCESS_NAME	VARCHAR2(8)		The name of the process executing the schedule. NULL if not currently executing

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		The session ID (SID, SERIAL#) of the job executing this schedule. NULL if not currently executing
INSTANCE	NUMBER		The OPS instance number executing the schedule
LAST_RUN_DATE	DATE		The date on the last successful execution
LAST_RUN_TIME	VARCHAR2(8)		The time of the last successful execution in HH:MI:SS format
CURRENT_START_ DATE	DATE		The date at which the current window of this schedule was started
CURRENT_START_TIME	VARCHAR2(8)		The time of day at which the current window of this schedule was started in HH:MI:SS format
NEXT_RUN_DATE	DATE		The date at which the next window of this schedule will be started
NEXT_RUN_TIME	VARCHAR2(8)		The time of day at which the next window of this schedule will be started in HH:MI:SS format
TOTAL_TIME	NUMBER		The total time, in seconds, spent by the system in executing this schedule
TOTAL_NUMBER	NUMBER		The total number of messages propagated in this schedule
TOTAL_BYTES	NUMBER		The total number of bytes propagated in this schedule
MAX_NUMBER	NUMBER		The maximum number of messages propagated in a propagation window
MAX_BYTES	NUMBER		The maximum number of bytes propagated in a propagation window
AVG_NUMBER	NUMBER		The average number of messages propagated in a propagation window
AVG_SIZE	NUMBER		The average size of a propagated message in bytes
AVG_TIME	NUMBER		The average time, in seconds, to propagate a message
FAILURES	NUMBER		The number of times the execution failed. If 16, the schedule will be disabled
LAST_ERROR_DATE	DATE		The date of the last unsuccessful execution
LAST_ERROR_TIME	VARCHAR2(8)		The time of the last unsuccessful execution
LAST_ERROR_MSG	VARCHAR2(4000)		The error number and error message text of the last unsuccessful execution

DBA_ QUEUE_TABLES

This view describes the names and types of the queues in all of the queue tables created in the database. For more information about this view and Advanced Queuing, see the *Oracle8i Application Developer's Guide - Fundamentals*.

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
COMPATIBLE	VARCHAR2(5)		The lowest release level which this is compatible with (eg, 8.0.4)
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		The instance number of the instance which currently owns the queue table
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user

DBA_ QUEUES

This view describes the operational characteristics for every queue in a database. For more information about this view and Advanced Queuing, see the *Oracle8i Application Developer's Guide - Fundamentals*.

Column	Datatypes	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the queue schema
NAME	VARCHAR2(30)	NOT NULL	Name of the queue

Column	Datatypes	NULL	Description
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	The name of the queue table where this queue resides
QID	NUMBER	NOT NULL	Unique queue identifier
QUEUE_TYPE	VARCHAR2(15)		Queue type: NORMAL_QUEUE - Normal queue EXCEPTION_QUEUE - Exception queue NON_PERSISTENT_QUEUE - Non-persistent queue
MAX_RETRIES	NUMBER		Number of dequeue attempts allowed
RETRY_DELAY	NUMBER		Time lapse in seconds before retry takes place
ENQUEUE_ENABLED	VARCHAR2(7)		YES / NO
DEQUEUE_ENABLED	VARCHAR2(7)		YES / NO
RETENTION	VARCHAR2(40)		Number of seconds message is retained after dequeue FOREVER - messages stay in the queue permanently
USER_COMMENT	VARCHAR2(50)		User comment about the table

DBA_RCHILD

This view 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

This view lists all the refresh groups.

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 or N; if Y, then destroy the refresh group when its last item is removed

Column	Datatype	NULL	Description
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		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 or 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=quick purge option; 2=precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap.

DBA_REFRESH_CHILDREN

This view lists all of the objects in refresh groups.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y 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

Column	Datatype	NULL	Description
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 or 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=quick purge option; 2=precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

DBA_REFS

This view describes the REF columns and REF attributes in object type columns of all the tables in the database.

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)		Is the REF value stored with ROWID? (YES or NO)
IS_SCOPED	VARCHAR2(3)		Is the REF column scoped? (YES or NO)
SCOPE_TABLE _OWNER	VARCHAR2(30)		Name of the owner of the scope table, if it exists
SCOPE_TABLE _NAME	VARCHAR2(30)		Name of the scope table, if it exists
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
HAS_REFERENTIAL _CONS	VARCHAR2(3)		Yes or no for a referential constraint
REFERENTIAL_CONS _NAME	VARCHAR2(30)		The name associated with the referential integrity constraint

DBA_REGISTERED_SNAPSHOT_GROUPS

This view lists all the snapshot repgroups at this site.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the snapshot replication group
SNAPSHOT_SITE	VARCHAR2(128)		Site of the master of the snapshot repgroup
GROUP_COMMENT	VARCHAR2(80)		Description of the snapshot repgroup
VERSION	VARCHAR2(8)		Version of the snapshot repgroup
FNAME	VARCHAR2(30)		Name of the flavor of the snapshot object group

DBA_REGISTERED_SNAPSHOTS

This view is used to get information about remote snapshots of local tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot
SNAPSHOT_SITE	VARCHAR2(128)	NOT NULL	Global name of the snapshot site
CAN_USE_LOG	VARCHAR2(3)		If set to NO, this snapshot is complex and cannot fast refresh
UPDATABLE	VARCHAR2(3)		If set to NO, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Values used to drive a refresh of the snapshot (PRIMARY KEY / ROWID / COMPLEX). If PRIMARY KEY, then the snapshot uses primary keys to drive a fast refresh. If ROWID, then it uses RowIDs to drive a fast refresh. If COMPLEX, then fast refreshes are not allowed and the snapshot can only perform complete refreshes
SNAPSHOT_ID	NUMBER(38)		Identifier for the snapshot used by the master for fast refresh
VERSION	VARCHAR2(17)		Version of snapshot
QUERY_TXT	LONG		Query defining the snapshot

DBA_REPCATLOG

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_REFRESH_TEMPLATES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_TEMPLATE_OBJECTS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_TEMPLATE_PARMS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_TEMPLATE_SITES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_USER_AUTHORIZATIONS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCAT_USER_PARM_VALUES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCOLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCOLUMN_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPCONFLICT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPDDL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPGENERATED

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPGENOBJECTS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPGROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPGROUPED_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPKEY_COLUMNS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPOBJECT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPPARAMETER_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPPRIORITY

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPPRIORITY_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPPROP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPRESOLUTION

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPRESOLUTION_METHOD

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPRESOL_STATS_CONTROL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_REPSITES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DBA_RGROUP

This view lists all refresh groups.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is removed
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N; if Y, then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		Y or N; if Y, proceed with refresh despite error when pushing deferred RPC’s
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER	NOT NULL	Identifier of job used to refresh the group automatically
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1=quick purge option; 2=precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

DBA_ROLES

This view 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_ROLE_PRIVS

This view 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_ROLLBACK_SEGS

This view contains descriptions of 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 parallel server instance number
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the segment header

DBA_RSRC_CONSUMER_GROUP_PRIVS

This view lists all resource consumer groups and the users and roles to which they have been granted.

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)		Whether grant was with the GRANT option
INITIAL_GROUP	VARCHAR2(3)		Whether consumer group is designated as the default

DBA_RSRC_CONSUMER_GROUPS

This view lists all resource consumer groups which exist in the database.

Column	Datatype	NULL	Description
CONSUMER_GROUP	VARCHAR2(30)		The consumer group name
CPU_METHOD	VARCHAR2(30)		The CPU resource allocation method for the consumer group
COMMENTS	VARCHAR2(2000)		A text comment on the consumer group
STATUS	VARCHAR2(30)		PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)		Whether the consumer group is mandatory

DBA_RSRC_MANAGER_SYSTEM_PRIVS

This view lists all the users and roles which have been granted system privileges pertaining to the resource manager.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	The name of the system privilege
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option

DBA_RSRC_PLANS

This view lists all resource plans which exist in the database.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		Plan name
NUM_PLAN_DIRECTIVES	NUMBER		The number of plan directives for the plan
CPU_METHOD	VARCHAR2(30)		The CPU resource allocation method for the plan
MAX_ACTIVE_SESS_TARGET_MTH	VARCHAR2(30)		RESERVED FOR FUTURE USE
PARALLEL_DEGREE_LIMIT_MTH	VARCHAR2(30)		The parallel degree limit resource allocation method for the plan
COMMENTS	VARCHAR2(2000)		A text comment on the plan
STATUS	VARCHAR2(30)		PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)		Whether the plan is mandatory

DBA_RSRC_PLAN_DIRECTIVES

This view lists all resource plan directives which exist in the database.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		The name of the plan to which this directive belongs
GROUP_OR_SUBPLAN	VARCHAR2(30)		The name of the consumer group / sub-plan referred to
TYPE	VARCHAR2(14)		Whether GROUP_OR_SUBPLAN refers to a consumer group or plan
CPU_P1	NUMBER		The first parameter for the CPU resource allocation method
CPU_P2	NUMBER		The second parameter for the CPU resource allocation method
CPU_P3	NUMBER		The third parameter for the CPU resource allocation method
CPU_P4	NUMBER		The fourth parameter for the CPU resource allocation method
CPU_P5	NUMBER		The fifth parameter for the CPU resource allocation method
CPU_P6	NUMBER		The sixth parameter for the CPU resource allocation method
CPU_P7	NUMBER		The seventh parameter for the CPU resource allocation method

Column	Datatype	NULL	Description
CPU_P8	NUMBER		The eighth parameter for the CPU resource allocation method
MAX_ACTIVE_SESS_TARGET_P1	NUMBER		RESERVED FOR FUTURE USE
PARALLEL_DEGREE_LIMIT_P1	NUMBER		The first parameter for the parallel degree limit resource allocation method
COMMENTS	VARCHAR2(2000)		A text comment on the plan directive
STATUS	VARCHAR2(30)		PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)		Whether the plan is mandatory

DBA_RULESETS

This view lists information about rulesets.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The user owning the ruleset
RULESET_NAME	VARCHAR2(30)	NOT NULL	The name of the ruleset
RULESET_STORAGE_TABLE	VARCHAR2(61)		The table in which the ruleset is stored (will be schema.table)
BASE_TABLE	VARCHAR2(61)		The table on which the rules are defined (will be schema.table)

DBA_SEGMENTS

This view contains information about storage allocated for all database segments.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the segment owner
SEGMENT_NAME	VARCHAR2(81)		Name, if any, of the segment
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(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

Column	Datatype	NULL	Description
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 of the initial extent of the segment
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)		Name of the default buffer pool for the appropriate object

DBA_SEQUENCES

This view contains descriptions of all sequences in the database.

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

DBA_SNAPSHOT_LOGS

This view lists all snapshot logs in the database.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot log
MASTER	VARCHAR2(30)	NOT NULL	Name of the master table of which the log logs changes
LOG_TABLE	VARCHAR2(30)	NOT NULL	Log table; holds timestamps and changes made to the master table
LOG_TRIGGER	VARCHAR2(30)		Obsolete with the release of Oracle8*i*. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserts rows into the log
FILTER_COLUMNS	VARCHAR2(3)		If set to YES, the snapshot log records filter column information
ROWIDS	VARCHAR2(3)		If set to YES, the snapshot log records ROWID information
PRIMARY_KEY	VARCHAR2(3)		If set to YES, the snapshot log records primary key information
CURRENT_SNAPSHOTS	DATE		One date per snapshot; the date the snapshot of the master last refreshed
SNAPSHOT_ID	NUMBER(38)		Unique identifier of the snapshot

DBA_SNAPSHOT_LOG_FILTER_COLS

This view lists all filter columns (excluding PK cols) being logged in the snapshot logs.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the master table being logged
NAME	VARCHAR2(30)	NOT NULL	Name of the master table being logged
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Filter column being logged

DBA_SNAPSHOT_REFRESH_TIMES

This view lists snapshot refresh times.

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

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

DBA_SNAPSHOTS

This view lists all snapshots in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	The view used by users and applications for viewing the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table the snapshot is stored in
MASTER_VIEW	VARCHAR2(30)		View of the master table, owned by the snapshot owner, used for refreshes. This is obsolete in Oracle8*i* and is set to NULL
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table of which this snapshot is a copy
MASTER_LINK	VARCHAR2(128)		Database link name to the master site
CAN_USE_LOG	VARCHAR2(3)		If NO, this snapshot is complex and will never use a log
UPDATABLE	VARCHAR2(3)		If NO, the snapshot is read only
LAST_REFRESH	DATE		SYSDATE from the master site at the time of the last refresh
ERROR	NUMBER		The number of failed automatic refreshes since last successful refresh
TYPE	VARCHAR2(8)		The type of refresh (complete, fast, force) for all automatic refreshes
NEXT	VARCHAR2(200)		The date function used to compute next refresh dates
START_WITH	DATE		The date expression for the first automatic refresh time.
REFRESH_GROUP	NUMBER		All snapshots in a given refresh group get refreshed in the same transaction
REFRESH_METHOD	VARCHAR2(11)		Values used to drive a fast refresh of the snapshot

Column	Datatype	NULL	Description
UPDATE_TRIG	VARCHAR2(30)		Obsolete with the release of Oracle8*i*. Set to NULL. Formerly, this parameter was the name of the trigger that fills the UPDATE_LOG
UPDATE_LOG	VARCHAR2(30)		The table that logs changes made to an updatable snapshots
QUERY	LONG		The original query of which this snapshot is an instantiation
FR_OPERATIONS	VARCHAR2(10)		Status of generated fast refresh operations: (REGENERATE, VALID)
CR_OPERATIONS	VARCHAR2(10)		Status of generated complete refresh operations: (REGENERATE, VALID)
MASTER_ROLLBACK_SEG	VARCHAR2(30)		The rollback segment used at the master site
STATUS	VARCHAR2(7)		The status of the contents of the snapshot
REFRESH_MODE	VARCHAR2(8)		This indicates how and when the snapshot will be refreshed
PREBUILT	VARCHAR2(3)		If YES, this snapshot uses a prebuilt table as the base table

DBA_SOURCE

This view contains source of all stored objects in the database.

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: PROCEDURE, FUNCTION, PACKAGE, TYPE, TYPE BODY, or PACKAGE BODY
LINE	NUMBER	NOT NULL	Line number of this line of source
TEXT	VARCHAR2(4000)		Source text

DBA_STMT_AUDIT_OPTS

This view contains information which describes current system auditing options across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client. NULL if the client is performing the operation directly
AUDIT_OPTION	VARCHAR2(40)	NOT NULL	Name of the system auditing option
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_SUBPART_COL_STATISTICS

This view lists column statistics and histogram information for table subpartitions.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying subpartitions?

Column	Datatype	NULL	Description
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

DBA_SUBPART_HISTOGRAMS

This view lists the actual histogram data (end-points per histogram) for histograms on table subpartitions.

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

DBA_SUBPART_KEY_COLUMNS

This view lists subpartitioning key columns for tables (and Local indexes on tables) partitioned using the Composite Range/Hash method.

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

DBA_SYNONYMS

This view lists all synonyms in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Username of the 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 in a remote synonym

DBA_SYS_PRIVS

This view lists system privileges granted to users and roles.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Grantee name, user, or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege
ADMIN_OPTION	VARCHAR2(3)		Grant was with the ADMIN option

DBA_TAB_COL_STATISTICS

This view contains column statistics and histogram information which is in the DBA_TAB_COLUMNS view. For more information, see "DBA_TAB_COLUMNS" on page 2-145.

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
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column

Column	Datatype	NULL	Description
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

DBA_TAB_COLUMNS

This view contains information which describes columns of all tables, views, and clusters. To gather statistics for this view, use the SQL command ANALYZE.

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
NULLABLE	VARCHAR2(1)		Does column allow NULL values?
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.
LOW_VALUE	RAW(32)		
HIGH_VALUE	RAW(32)		
DENSITY	NUMBER		
NUM_NULLS	NUMBER		

Column	Datatype	NULL	Description
NUM_BUCKETS	NUMBER		The number of buckets in histogram for the column
LAST_ANALYZED	DATE		The date of the most recent time this column was analyzed
SAMPLE_SIZE	SAMPLE_SIZE		The sample size used in analyzing this column
CHARACTER_SET _NAME	VARCHAR2(44)		The name of the character set: CHAR_CS, NCHAR_ CS
CHAR_COL_DECL _LENGTH	NUMBER		Declaration length of character type column
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		Average column length in bytes

DBA_TAB_COMMENTS

This view contains comments on all tables and views in the database.

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

DBA_TAB_HISTOGRAMS

This view lists histograms on columns of all tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of Table
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
ENDPOINT_NUMBER	NUMBER		Endpoint number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

DBA_TAB_PARTITIONS

This view describes, for each table partition, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COMPOSITE	VARCHAR2(3)		’YES’ if the partition belongs to a Local index on a table partitioned using Composite method; ’NO’ otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
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

Column	Datatype	NULL	Description
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 of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)	NOT NULL	The buffer pool for the partition
GLOBAL_STATS	VARCHAR2(3)		The global statistics
USER_STATS	VARCHAR2(3)		The user statistics

DBA_TAB_PRIVS

This view lists all grants on objects in the database.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	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	Table Privilege
GRANTABLE	VARCHAR2(3)		Privilege is grantable

DBA_TAB_SUBPARTITIONS

This view describes, for each table subpartition, its name, name of the table and partition to which it belongs, and its storage attributes. Note that statistics will not be 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

Column	Datatype	NULL	Description
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 when last analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

DBA_TABLES

This view contains descriptions of all relational tables in the database. To gather statistics for this view, use the SQL command ANALYZE.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned 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
PCT_FREE	NUMBER		Minimum percentage of free space in a block; NULL for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; NULL for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; NULL for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; NULL for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes; NULL for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes; NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; NULL for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; NULL for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to this segment; NULL for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment; NULL for partitioned tables
LOGGING	VARCHAR2(3)		Whether logging is enabled (YES or NO); NULL for partitioned tables
BACKED_UP	VARCHAR2(1)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		Number of rows returned by the ANALYZE command

Column	Datatype	NULL	Description
BLOCKS	NUMBER		The number of blocks below the high water mark
EMPTY_BLOCKS	NUMBER		The number of empty (never used) data blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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)		Number of query servers used for a full-table scan
INSTANCES	VARCHAR2(10)		The number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Whether the table is to be cached in the buffer cache
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 of the most recent time this table was 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 or IOT_OVERFLOW. If this is not an index organized table, then IOT_TYPE is NULL
TEMPORARY	VARCHAR2(1)		Whether the table is temporary (Y or N)
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object; NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

DBA_TABLESPACES

This view contains descriptions of all tablespaces.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
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	NOT NULL	Default maximum number of extents
PCT_INCREASE	NUMBER	NOT NULL	Default percent increase for extent size
MIN_EXTLEN	NUMBER		Minimum extent size for the tablespace
STATUS	VARCHAR2(9)		Tablespace status: ONLINE, OFFLINE, or READ ONLY
CONTENTS	VARCHAR2(9)		Tablespace contents: "PERMANENT", or "TEMPORARY"
LOGGING	VARCHAR2(9)		Default logging attribute
EXTENT_ MANAGEMENT	VARCHAR2(10)		Extent management tracking: "DICTIONARY" or "LOCAL"
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for this tablespace
PLUGGED_IN	VARCHAR2(3)		YES - the tablespace is plugged in; NO - it is not plugged in

DBA_TEMP_FILES

This view contains information about database temp files.

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" or "NO"
MAXBYTES	NUMBER		maximum size of the file in bytes

Column	Datatype	NULL	Description
MAXBLOCKS	NUMBER		Maximum size of the file in ORACLE blocks
INCREMENT_BY	NUMBER		Default increment for autoextension
USER_BYTES	NUMBER		Size of the useful portion of file in bytes
USER_BLOCKS	NUMBER		Size of the useful portion of file in ORACLE blocks

DBA_TRIGGERS

This view lists all triggers in the database.

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(75)		Events that fire the trigger: INSERT, UPDATE, DELETE, STARTUP, SHUTDOWN, ERROR, LOGON, LOGOFF, CREATE, ALTER, DROP
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
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)		WHEN clause. Must evaluate to TRUE for TRIGGER_ BODY to execute.
STATUS	VARCHAR2(8)		Whether the trigger is enabled: ENABLED or 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

DBA_TRIGGER_COLS

This view lists column usage in all triggers.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)	NOT NULL	Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)	NOT NULL	Name of the trigger
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column used in trigger definition
COLUMN_LIST	VARCHAR2(3)		Is column specified in UPDATE OF clause?
COLUMN_USAGE	VARCHAR2(17)		Usage of column within trigger body

DBA_TS_QUOTAS

This view lists tablespace quotas for all users.

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

This view displays the attributes of types in the database.

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

Column	Datatype	NULL	Description
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute or maximum length of the VARCHAR or VARCHAR2 attribute
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute or binary precision of the FLOAT attribute
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET _NAME	VARCHAR2(44)		The name of the character set: 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 ID number)

DBA_TYPE_METHODS

This view is a description of methods of all types in the database.

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 the ID number
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method

DBA_TYPES

This view displays all abstract datatypes in the database.

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

Column	Datatype	NULL	Description
ATTRIBUTES	NUMBER		Number of attributes in the type
METHODS	NUMBER		Number of methods in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type

DBA_UNUSED_COL_TABS

This view contains a description of all tables containing unused columns.

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

DBA_UPDATABLE_COLUMNS

This view contains a description of columns that are updatable by the database administrator in a join view. See *Oracle8i Concepts* for information on updatable join views.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

DBA_USERS

This view lists information about all users of the database.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user

Column	Datatype	NULL	Description
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
EXTERNAL_NAME	VARCHAR2(4000)		User external name

DBA_USTATS

This view contains information about 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
PROPERTY	VARCHAR2(6)		Property of the object - column or index
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

DBA_VARRAYS

This view lists the text of views accessible to the user.

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

DBA_VIEWS

This view contains the text of all views in the database.

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

DBA_WAITERS

This view shows all the sessions that have someone waiting on a lock they hold, but that are not themselves waiting on a lock.

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

This view lists 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

This view lists 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)

DEFCALL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFCALLDEST

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFDEFAULTDEST

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFERRCOUNT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFERROR

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFLOB

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFPROPAGATOR

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFSCHEDULE

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFTRAN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

DEFTRANDEST

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

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 2-171.)

DICT

This is a synonym for DICTIONARY. For more information, see "DICTIONARY" on page 2-161.

DICTIONARY

This view 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

DICT_COLUMNS

This view 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

ERROR_SIZE

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" on page 2-110 and "USER_OBJECT_SIZE" on page 2-224.

EXCEPTIONS

This view 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 is a Parallel Server view. This view displays the mapping of PCM locks to datafiles as specified in initialization parameter GC_FILES_TO_LOCKS. For more information on this parameter, see "GC_FILES_TO_LOCKS" on page 1-44.

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

Column	Datatype	NULL	Description
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 blocks protected by a PCM lock on the datafile

For more information on this view, see *Oracle8i Parallel Server Concepts and Administration*.

FILE_PING

This is a Parallel Server view. This 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. For more information on this parameter, see "GC_FILES_TO_LOCKS" on page 1-44.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

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
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

Column	Datatype	NULL	Description
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
N_2_SS	NUMBER		Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file

FILEXT$

This view is the equivalent of DBA_DATA_FILES. Oracle recommends you use DBA_DATA_FILES instead of FILEXT$. For more information, see "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

This view 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

This view 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

This view contains data dictionary information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)		Data dictionary table name

Column	Datatype	NULL	Description
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

This view 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= this is an environment variable, F= 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

This view 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

This view displays information from the base data dictionary translation table.

Column	Datatype	NULL	Description
DD_TABLE_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

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

HS_CLASS_CAPS

This view 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 which is incremented for every row inserted (used internally)

HS_CLASS_DD

This view 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
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which 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 which is incremented for every row inserted (used internally)

HS_CLASS_INIT

This view 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= this is an environment variable, F= 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 which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_EXTERNAL_OBJECT_PRIVILEGES

This view 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. Name is 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

This view 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. Name is unique for each instance

Column	Datatype	NULL	Description
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

This view 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

This view contains information about legal non-Oracle (FDS) classes.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (eg, ODBC, DB2)
FDS_CLASS _COMMENTS	VARCHAR2(255)		Text description of the non-Oracle class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_FDS_INST

This view 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 which is incremented for every row inserted (used internally)

Column	Datatype	NULL	Description
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_CAPS

This view 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 (eg, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_DD

This view 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 (eg, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
DD_TABLE_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

Column	Datatype	NULL	Description
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_INIT

This view 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= this is an environment variable, F= 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 which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which 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

This is a synonym for USER_INDEXES. For more information, see "USER_INDEXES".

INDEX_HISTOGRAM

This view contains information from the ANALYZE INDEX ... VALIDATE STRUCTURE command.

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

This view stores information from the last ANALYZE INDEX ... VALIDATE STRUCTURE command.

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
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

Column	Datatype	NULL	Description
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

NLS_DATABASE_PARAMETERS

This view lists permanent NLS parameters of the database.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)	NOT NULL	Parameter name
VALUE	VARCHAR2(30)		Parameter value

NLS_INSTANCE_PARAMETERS

This view lists NLS parameters of the instance.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(30)		Parameter value

NLS_SESSION_PARAMETERS

This view lists NLS parameters of the user session.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(30)		Parameter value

OBJ

This is a synonym for USER_OBJECTS. For more information, see "USER_OBJECTS".

ORPHAN_KEY_TABLE

If there are index entries that point to corrupt rows when you run the DUMP_ORPHAN_KEYS procedure, the orphan key table is populated as a log of the activity and provides index information.

Column	Datatype	Null	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
IPART_NAME	VARCHAR2(30)	NULL	Index partition / subpartition name
INDEX_ID	NUMBER	NOT NULL	Index’s object id
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PART_NAME	VARCHAR2(30)	NULL	Table partition / subpartition name
TABLE_ID	NUMBER	NOT NULL	Base table’s object id
KEYDATA	UROWID	NOT NULL	Index entry’s keydata
KEY	UROWID	NOT NULL	Index entry’s key in linear format
DUMP_TIMESTAMP	DATE	NOT NULL	Timestamp

PARSED_PIECES

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" and "USER_OBJECT_SIZE".

PARSED_SIZE

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" and "USER_OBJECT_SIZE".

PLAN_TABLE

This view 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(30)		Options used for the operation performed at this step
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_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		Not currently used
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		Holds SQL text for remote cursors and parallel execution slaves
DISTRIBUTION	VARCHAR2(30)		The distribution method

PLUGGABLE_SET_CHECK

This view 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

Column	Datatype	NULL	Description
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

This view 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

This view describes users who can assume the identity of other users.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of the client whose identity a proxy may assume
CLIENT	VARCHAR2(30)	NOT NULL	Name of a user which assume the identity of a client
ROLE	VARCHAR2(30)		Name of a role that a proxy user may or may not activate while acting as a client
FLAGS	VARCHAR2(35)		Indicates whether or not a proxy can activate all client roles, no client roles, or a specific client role

PSTUBTBL

This table contains information on stubs generated by the PSTUB utility so that an Oracle Forms 3.0 client can call stored procedures in 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

PUBLICSYN

This view 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

PUBLIC_DEPENDENCY

This view 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)

QUEUE_PRIVILEGES

This view 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

RC

RC_ views are not created by default in the database, they are only available after you create an optional recovery catalog (which contains schemas containing information about backups) for use with Recovery Manager. Please see the *Oracle8i Backup and Recovery Guide* for details about these views.

REPAIR_TABLE

The REPAIR_TABLE view lists any corruptions found by the CHECK_OBJECT procedure. It also displays information about how the corruptions would be addressed if the FIX_CORRUPT_BLOCKS procedure is executed. This information is also used by the FIX_CORRUPT_BLOCKS and REFORMAT_CORRUPT_BLOCKS procedures on execution.

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Object number
TABLESPACE_ID	NUMBER	NOT NULL	Tablespace number
RELATIVE_FILE_ID	NUMBER)	NOT NULL	Relative file number
BLOCK_ID	NUMBER	NOT NULL	Block number
CORRUPT_TYPE	NUMBER	NOT NULL	Type of corruption encountered
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Object name

Column	Datatype	NULL	Description
BASEOBJECT_NAME	VARCHAR2(30)	NULL	Base table name when object is an index
PARTITION_NAME	VARCHAR2(30)	NULL	Partition or subpartition name
CORRUPT_DESCRIPTION	VARCHAR2(200)	NULL	Description of corruption
REPAIR_DESCRIPTION	VARCHAR2(200)	NULL	Description of repair action
MARKED_CORRUPT	VARCHAR2(10)	NOT NULL	Indicates whether the block is marked corrupt: ’TRUE’ or ’FALSE’
CHECK_TIMESTAMP	DATE	NOT NULL	Timestamp from check_object procedure
FIX_TIMESTAMP	DATE	NULL	Set if block is modified by fix_ corrupt_blocks
REFORMAT_TIMESTAMP	DATE	NULL	Set if block is modified by reformat_ corrupt_blocks

RESOURCE_COST

This view 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

This view contains descriptions for resources. It maps the resource name to the resource number.

Column	Datatype	NULL	Description
RESOURCE#	NUMBER	NOT NULL	Numeric resource code
TYPE#	NUMBER	NOT NULL	Name of type
NAME	VARCHAR2(32)	NOT NULL	Name of resource

ROLE_ROLE_PRIVS

This view contains information about roles granted to other roles. (Information is only provided 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

This view 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

This view 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
GRANTABLE	VARCHAR2(3)		YES if the role was granted with ADMIN OPTION; otherwise NO

SEQ

This is a synonym for USER_SEQUENCES. For more information see "USER_SEQUENCES" on page 2-243.

SESSION_CONTEXT

This view lists set attributes information for the current session.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)		The namespace that the active attribute is in
ATTRIBUTE	VARCHAR2(30)		The name of the active attribute
VALUE	VARCHAR2(64)		The value of the active attribute

SESSION_PRIVS

This view 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

This view 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

This view is accessed to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views. For more information, see "DBA_OBJECT_SIZE" on page 2-110, and "USER_OBJECT_SIZE" on page 2-224.

STMT_AUDIT_OPTION_MAP

This view contains information about 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

This is a synonym for USER_SYNONYMS. For more information, see "USER_SYNONYMS" on page 2-247.

SYNONYMS

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSCATALOG

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSFILES

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSSEGOBJ

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSTEM_PRIVILEGE_MAP

This view contains information about system privilege codes.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege type code

Column	Datatype	NULL	Description
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege
PROPERTY	NUMBER	NOT NULL	

SYS_OBJECTS

This view 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

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

TABLE_PRIVILEGES

This view 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. Use of this view is not recommended.

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

Column	Datatype	NULL	Description
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

TABLE_PRIVILEGE_MAP

This view 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)

TABS

This is a synonym for USER_TABLES. For more information, see "USER_TABLES" on page 2-254.

TABQUOTAS

This view is included for compatibility with Oracle version 5. Use of this view is not recommended.

TRUSTED_SERVERS

This view 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. See the examples below
NAME	VARCHAR2(128)		Server name. Can be a specific server name or "ALL" for all servers

For example:

If all servers are trusted, then TRUSTED_SERVERS returns:

```
TRUST        NAME
-------      ------
Trusted      ALL
```

If none of the servers are trusted, then TRUSTED_SERVERS returns:

```
TRUST        NAME
-------      ------
Untrusted      ALL
```

If all servers are trusted, except DB1, then TRUSTED_SERVERS returns:

```
TRUST        NAME
-------      ------
Untrusted     DB1
```

If all servers are untrusted, except DB1, then TRUSTED_SERVERS returns:

```
TRUST        NAME
-------      ------
Trusted      DB1
```

For more information on this view see *Oracle8i Distributed Database Systems*.

TS_PITR_CHECK

This view, created by CATPITR.SQL provides information on any dependencies or restrictions which might prevent tablespace point-in-time recovery from proceeding. This view applies only to the tablespace point-in-time recovery feature. For more information, see *Oracle8i 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

This view 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

This view contains pluggable check information.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name

Column	Datatype	NULL	Description
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

This table contains descriptions of the tables (object tables and relational tables) available to the user.

Column	Datatype	NULL	Description
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

Column	Datatype	NULL	Description
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If index organized table, then IOT_TYPE is IOT or IOT_ OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)		Type of the table if the table is a typed table
PACKED	VARCHAR2(1)		If the table is a typed table, does it store objects in packed format?
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2(1)		Is the index object created as part of icreate for domain indexes?
NESTED	VARCHAR2(3)		Is the table a nested table?

Column	Datatype	NULL	Description
BUFFER_POOL	VARCHAR2(7)		The default buffer pool to be used for table blocks
ROW_MOVEMENT	VARCHAR2(8)		The movement of the row
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

USER_ARGUMENTS

This view lists the arguments in the object which are accessible to the user.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)		Procedure or function name
PACKAGE_NAME	VARCHAR2(30)		Package name
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
OVERLOAD	VARCHAR2(40)		Overload unique identifier
ARGUMENT_NAME	VARCHAR2(30)		Argument name
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: 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

Column	Datatype	NULL	Description
TYPE_OWNER	VARCHAR2(30)		Owner name of the type
TYPE_NAME	VARCHAR2(30)		Name
TYPE_SUBNAME	VARCHAR2(30)		This is valid only in case of package local types; in such cases, the package name is the name and the type name is the subname
TYPE_LINK	VARCHAR2(128)		Database link valid only in case of package local types, in case the package is remote

USER_ASSOCIATIONS

This view lists user-defined statistics information

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	Object name 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)		Schema type of the object - column, type, package or function
STATSTYPE_ SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of Statistics type which contains the cost, selectivity or stats funcs
DEF_SELECTIVITY	NUMBER		Default Selectivity if any of the object
DEF_CPU_COST	NUMBER		Default CPU cost if any of the object
DEF_IO_COST	NUMBER		Default I/O cost if any of the object
DEF_NET_COST	NUMBER		Default Networking cost if any of the object

USER_AUDIT_OBJECT

This view, created by CATAUDIT.SQL, lists audit trail records for statements concerning objects.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier for 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 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 ACTION
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object renamed by a RENAME statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 11 characters, one for each action type, in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, and UPDATE; coded: for none, S for success, F for failure, and B for both)
COMMENT_TEXT	VARCHAR2(4000)		Text Comment on the audit trail entry (inserted by an application program)
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 (a statement may cause many actions)
RETURNCODE	NUMBER	NOT NULL	Oracle message code generated by the action (zero if the action succeeded)
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

USER_AUDIT_SESSION

This view, created by CATAUDIT.SQL, lists all audit trail records concerning connections and disconnections for the user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system log on user name of the user whose actions were audited

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier for 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 ACTION
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 message code generated by the action (zero if the action succeeded)

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.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system log on 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 for 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 object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action

Column	Datatype	NULL	Description
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in ACTION
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted/revoked by a GRANT/REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted/revoked by a GRANT/REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with ADMIN option
GRANTEE	VARCHAR2(30)		Username of the grantee specified in a GRANT/REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 11 characters, one for each action type, in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, and UPDATE; coded: for none, S for success, F for failure, and B for both)
COMMENT_TEXT	VARCHAR2(4000)		Text Comment on the audit trail entry (inserted by an application program)
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 (a statement may cause many actions)
RETURNCODE	NUMBER	NOT NULL	Oracle message code generated by the action (zero if the action succeeded)
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

USER_AUDIT_TRAIL

This view, created by CATAUDIT.SQL, lists audit trail entries relevant to the user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system log on 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

Column	Datatype	NULL	Description
TERMINAL	VARCHAR2(2000)		Identifier for 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 object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of object affected by the action
ACTION	NUMBER	NOT NULL	Numeric type code corresponding to the action name
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in ACTION
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object renamed by a RENAME statement
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted / revoked by a GRANT / REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted / revoked by a GRANT / REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with ADMIN option
GRANTEE	VARCHAR2(30)		Username of the grantee specified in a GRANT / 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

Column	Datatype	NULL	Description
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 (a statement can cause many actions)
RETURNCODE	NUMBER	NOT NULL	Oracle message code generated by the action (zero if the action succeeded)
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

USER_CATALOG

This view lists tables, views, synonyms, and sequences owned by the user.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
TABLE_TYPE	VARCHAR2(11)		Type of the object

USER_CLUSTERS

This view contains descriptions of user’s own clusters.

Column	Datatype	NULL	Description
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

Column	Datatype	NULL	Description
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
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 hash 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)		The number of threads per instance for scanning the cluster
INSTANCES	VARCHAR2(10)		The 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
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
SINGLE_TABLE	VARCHAR2(5)		Y if the cluster is single table; N if not

USER_CLUSTER_HASH_EXPRESSIONS

This view list hash functions for the users’ hash clusters.

This view list all clusters accessible to the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of owner of cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of cluster
HASH_EXPRESSION	LONG		Text of hash function of cluster

USER_CLU_COLUMNS

This view contains a mapping of columns in user's tables to cluster columns.

Column	Datatype	NULL	Description
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

USER_COL_COMMENTS

This view lists comments on columns of user's tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Object name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
COMMENTS	VARCHAR2(4000)		Comment on the column

USER_COL_PRIVS

This view lists grants on columns for which the user is the owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

USER_COL_PRIVS_MADE

This view lists all grants on columns of objects owned by the user.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

USER_COL_PRIVS_RECD

This view lists grants on columns for which the user is the grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
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)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

USER_COLL_TYPES

This new data dictionary view displays the user’s named collection types.

Column	Datatype	NULL	Description
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
COLL_TYPE	VARCHAR2(30)	NOT NULL	Collection type
UPPER_BOUND	NUMBER		Maximum size of the VARYING ARRAY type
ELEM_TYPE_MOD	VARCHAR2(7)		Type modifier of the element
ELEM_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the element

Column	Datatype	NULL	Description
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the type of the element
LENGTH	NUMBER		Length of the CHAR element or maximum length of the VARCHAR or VARCHAR2 element
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL element or binary precision of the FLOAT element
SCALE	NUMBER		Scale of the NUMBER or DECIMAL element
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS, NCHAR_CS
ELEM_STORAGE	VARCHAR2(7)		Storage optimization specification for VARRAY of numeric elements
NULLS_STORED	VARCHAR2(3)		Is null information stored with each VARRAY element?

USER_CONSTRAINTS

This view lists constraint definitions on user's tables.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with 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 table with constraint definition
SEARCH_CONDITION	LONG		Text of search condition for table check
R_OWNER	VARCHAR2(30)		Owner of table used in referential constraint
R_CONSTRAINT_NAME	VARCHAR2(30)		Name of unique constraint definition for referenced table
DELETE_RULE	VARCHAR2(9)		The delete rule for a referential constraint: CASCADE, NO ACTION
STATUS	VARCHAR2(8)		Enforcement status of constraint: ENABLED or DISABLED

Column	Datatype	NULL	Description
DEFERRABLE	VARCHAR2(14)		Indicates whether the constraint is deferrable
DEFERRED	VARCHAR2(9)		Indicates whether the constraint was initially deferred
GENERATED	VARCHAR2(14)		Indicates whether the name system is generated
LAST_CHANGE	VARCHAR2(3)		Indicates when the constraint was last enabled or disabled
VALIDATED	DATE3		Indicates whether all data obeys the constraint: VALIDATED, NOT VALIDATED
RELY	VARCHAR2(4)		If set, this flag will be used in optimizer

USER_CONS_COLUMNS

This view contains information about columns in constraint definitions owned by the user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name associated with the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name associated with table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name associated with column or attribute of the object type column specified in the constraint definition
POSITION	NUMBER		Original position of column or attribute in definition

Note: If you create a constraint on a user-defined REF column, the system creates the constraint on the exploded columns that make up the REF column and hence column names of the constraint will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

USER_DB_LINKS

This view contains information on database links owned by the user.

Column	Datatype	NULL	Description
DB_LINK	VARCHAR2(128)	NOT NULL	Name of the database link
USERNAME	VARCHAR2(30)		Name of user to log in as
PASSWORD	VARCHAR2(30)		Password for login
HOST	VARCHAR2(2000)		Net8 string for connect
CREATED	DATE	NOT NULL	Creation time of the database link

USER_DEPENDENCIES

This view lists dependencies to and from a user's objects. Dependencies on views created without any database links are also available.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
REFERENCED_OWNER	VARCHAR2(64)		Owner of the parent object
REFERENCED_NAME	VARCHAR2(30)		Type of the parent object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
REFERENCED_TYPE	VARCHAR2(12)		Type of the parent object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
REFERENCED_LINK_NAME	VARCHAR2(128)		Name of the link to the parent object (if remote)
SCHEMAID	NUMBER		The schema ID
DEPENDENCY_TYPE	VARCHAR2(4)		Two values: REF when the dependency is a REF dependency; HARD otherwise

USER_DIMENSIONS

This view represents dimension objects.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension

Column	Datatype	NULL	Description
INVALID	VARCHAR2(1)		’Y’ if this dimension is in an invalid state, else ’N’
REVISION	NUMBER		Dimension revision level

USER_DIM_ATTRIBUTES

This view represents the relationship between a dimension level and a functionally dependent column. The table that the level columns are in must match the table of the dependent column.

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 this attribute is inferred from a JOIN KEY specification

USER_DIM_CHILD_OF

This view represents a 1:n hierarchical relationship between a pair of levels in a dimension.

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

USER_DIM_JOIN_KEY

This view represents a join between two dimension tables. The join is always specified between a parent dimension level column and a child column.

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	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 key column
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column

USER_DIM_HIERARCHIES

This view represents a dimension hierarchy.

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

USER_DIM_LEVELS

This view represents a dimension level. All columns of a dimension level must come from the same relation.

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

USER_DIM_LEVEL_KEY

This view represents a column of a dimension level. The position of a column within a level is specified by KEY_POSITION.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

USER_ERRORS

This view lists current errors on all a user’s stored objects.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY
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

USER_EXTENTS

This view lists extents of the segments belonging to a user’s objects.

Column	Datatype	NULL	Description
SEGMENT_NAME	VARCHAR2(81)		Name of the segment associated with the extent
SEGMENT_TYPE	VARCHAR2(17)		Type of the segment: INDEX PARTITION, TABLE PARTITION
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects).
EXTENT_ID	NUMBER	NOT NULL	Extent number in the segment
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER	NOT NULL	Size of the extent in Oracle blocks

USER_FREE_SPACE

This view lists the free extents in tablespaces accessible to the 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
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

USER_INDEXES

This view contains descriptions of the user’s own indexes. To gather statistics for this view, use the SQL command ANALYZE. This view supports parallel partitioned index scans.

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(12)		Type of index
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the indexed object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
TABLE_TYPE	VARCHAR2(11)		Type of the indexed object
UNIQUENESS	VARCHAR2(9)		Uniqueness status of the index: UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(11)		Enabled or disabled
PREFIX_LENGTH	NUMBER		The number of columns in the prefix of the key used for compression

Column	Datatype	NULL	Description
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 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
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		User column-id for last column to be included in index organized table top index
FREELISTS	NUMBER		Number of process freelists allocated in 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 attribute
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. This statistic is 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. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed column(s). This statistic is rounded to the nearest integer

Column	Datatype	NULL	Description
CLUSTERING_FACTOR	NUMBER		This statistic represents the amount of order of the rows in the table based on the values of the index. If its value is near the number of blocks, then the table is well ordered. In such a case, the index entries in a single leaf block tend to point to rows in the same data blocks. If its value is near the number of rows, then the table is randomly ordered. In such a case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks
STATUS	VARCHAR2(8)		State of the index: DIRECT LOAD, VALID or INPROGS (a DDL operation on the domain index is in progress).
DOMIDX_STATUS	VARCHAR		Reflects the status of the domain index. A NULL value means that the specified index is not a domain index. A Value of VALID means that the index is a domain index and the index does not have any errors. If the value of this column is IDXTYP_INVLD it means that the indextype corresponding to this domain index is invalid.
DOMIDX_OPSTATUS	VARCHAR		Reflects the status of an operation that was performed on the domain index. A value of NULL indicates that the specified index is not a domain index. A value of VALID specifies that the index does not have any errors. A value of FAILED indicates that the operation that was performed on the domain index failed with an error.
FUNCIDX_STATUS	VARCHAR		A value of NULL indicates a non function-based index. ENABLED indicates the function-based index is enabled. DISABLED indicates the function-based index is disabled.
NUM_ROWS	NUMBER		Number of rows in this index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze this index
LAST_ANALYZED	DATE		Timestamp for when this index was last analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index. NULL if PARTITIONED=NO
INSTANCES	VARCHAR2(40)		The number of instances across which the partitioned index is to be scanned
PARTITIONED	VARCHAR2(3)		Indicates whether this index is partitioned. Set to YES if it is partitioned
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this index system generated?
SECONDARY	VARCHAR2(1)		Is the index object created as part of icreate for domain indexes?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

Column	Datatype	NULL	Description
DURATION	VARCHAR2(15)		The duration
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

USER_IND_COLUMNS

This view lists columns of the user’s indexes or on user’s tables.

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column or attribute
DESCEND	VARCHAR2(4)		Y/N, Y if this column is sorted in descending order

Note: If you create an index on a user-defined REF column, the system creates the index on the exploded columns that make up the REF column and hence column names of the index will be the column names of the exploded columns. These names will however have as their prefix the REF column's name.

USER_IND_EXPRESSIONS

This view lists functional index expressions on user’s tables.

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Functional index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

USER_IND_PARTITIONS

This view describes, for each index partition that the current user owns, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
COMPOSITE	VARCHAR2(3)		’YES’ if the partition belongs to a Local index on a table partitioned using Composite method; ’NO’ otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the index
STATUS	VARCHAR2(8)		Indicates whether index partition is usable or not
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key

Column	Datatype	NULL	Description
AVG_DATA_BLOCKS _PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for
NUM_ROWS	NUMBER		Number of rows in this index partition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The buffer pool for the partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
PCT_DIRECT_ACCESS	NUMBER		If index on IOT, then this is percentage of rows with Valid guess

USER_IND_SUBPARTITIONS

This view describes, for each index subpartition that the current user owns, the partition level partitioning information, the storage parameters for the subpartition, and various partition statistics determined by ANALYZE.

Column	Datatype	NULL	Description
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index 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
STATUS	VARCHAR2(8)		Indicates 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

Column	Datatype	NULL	Description
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level
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 per key
AVG_DATA_BLOCKS _PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Measurement of the amount of (dis)order of the table this index partition is for
NUM_ROWS	NUMBER		Number of rows in this index partition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		The buffer pool for the partition
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

USER_INDEXTYPES

This view lists all the indextypes.

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
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of indextype implementation
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype

USER_INDEXTYPE_OPERATORS

This view lists all the operators supported by indextypes.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
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

USER_JOBS

This view lists all jobs owned by the user. For more information, see the *Oracle8i Administrator's Guide*.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change it
LOG_USER	VARCHAR2(30)	NOT NULL	USER who was logged in 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 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
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2(8)		Same as NEXT_DATE. The job becomes due for execution at this time
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job, in seconds
BROKEN	VARCHAR2(1)		If Y, no attempt is being made to run this job. See DBMS_JOB.BROKEN (JOB)

Column	Datatype	NULL	Description
INTERVAL	VARCHAR2(200)	NOT NULL	A date function, evaluated at the start of execution, becomes next NEXT_DATE
FAILURES	NUMBER		How many times has this job 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)		ALTER SESSION parameters describing the NLS environment of the job
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		Indicates which instance can execute the job; default is 0

USER_LIBRARIES

This view lists all the libraries that a user owns.

Column	Datatype	NULL	Description
LIBRARY_NAME	VARCHAR2(30)	NOT NULL	Library name
FILE_SPEC	VARCHAR2(2000)		File specification associated with the library
DYNAMIC	VARCHAR2(1)		Is the library dynamically loadable? (YES or NO)
STATUS	VARCHAR2(7)		Status of the library

USER_LOBS

This view displays the user’s LOBs contained in the user’s tables. It is only for internal LOBs, so BLOBs, CLOBs, and NCLOBs are OK. External LOBs (i.e.,BFILES) are not.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table containing the LOB
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB segment
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB index
CHUNK	NUMBER		Size of the LOB chunk as a unit of allocation/manipulation in bytes
PCTVERSION	NUMBER	NOT NULL	Maximum percentage of the LOB space used for versioning

Column	Datatype	NULL	Description
CACHE	VARCHAR2(3)		Indicates whether the LOB is accessed through the buffer cache
LOGGING	VARCHAR2(3)		Indicates whether the changes to the LOB are logged
IN_ROW	VARCHAR2(3)		Are some of the LOBs stored with the base row?

USER_LOB_PARTITIONS

This view displays the LOBs contained in tables accessible to the user.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a partitioned LOB item
PARTITION_NAME	VARCHAR2(30)		Name of a table partition
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the lOB item
COMPOSITE	VARCHAR2(3)		Is it a Composite partition (one of YES, NO)?
CHUNK	NUMBER		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size of the initial extent in bytes for a LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size of secondary extents in bytes for a LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for a LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of a LOB data partition

Column	Datatype	NULL	Description
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

USER_LOB_SUBPARTITIONS

This view displays partition-level attributes of LOB data subpartitions.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Name of a LOB column
LOB_NAME	VARCHAR2(30)		Name of a 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 a table subpartition to which this LOB subpartition corresponds
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		CHUNK attribute of a LOB data partition
PCTVERSION	NUMBER		PCTVERSION attribute of a LOB data partition
CACHE	VARCHAR2(3)		CACHE attribute of a LOB data partition
IN_ROW	VARCHAR2(3)		IN_ROW attribute of a LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size of the initial extent in bytes for a LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size of secondary extents in bytes for a LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of a LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of a LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for a LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of a LOB data partition

Column	Datatype	NULL	Description
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of a LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of a LOB data partition
BUFFER_POOL	VARCHAR2(7)		Buffer pool of a LOB data partition

USER_METHOD_PARAMS

This view is a description of method parameters of the user's types.

Column	Datatype	NULL	Description
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)
PARAM_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number or position
PARAM_MODE	VARCHAR2(6)		Mode of the parameter
PARAM_TYPE_MOD	VARCHAR2(7)		Type modifier of the parameter
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)		The name of the character set: CHAR_CS NCHAR_CS

USER_METHOD_RESULTS

This view is a description of method results of the user's types.

Column	Datatype	NULL	Description
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)
RESULT_TYPE_MOD	VARCHAR2(7)		Type modifier of the result
RESULT_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the result

Column	Datatype	NULL	Description
RESULT_TYPE_NAME	VARCHAR2(30)		Name of the type of the result
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS NCHAR_CS

USER_MVIEW_AGGREGATES

This view represents the grouping functions (aggregated measures) that appear in the SELECT list of an aggregated materialized view.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view index
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Position of this measure within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Set to Y if DISTINCT aggregation
MEASURE	LONG		Contains the SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*)

USER_MVIEW_ANALYSIS

This view represents the materialized views that potentially support query rewrite and that have additional information that is available for analysis by applications. This view excludes any materialized view that references a remote table or that includes a reference to a non-static value such as SYSDATE or USER. It also excludes any materialized view that was created as a snapshot prior to Oracle8.1 and that was 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
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view table object
CONTAINER_NAME	VARCHAR2(30)		Name of the snapshot that the materialized view data is contained in
LAST_REFRESH_SCN	NUMBER		The System Change Number

Column	Datatype	Null	Description
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		FORCE, FAST, COMPLETE, or NONE
SUMMARY	VARCHAR2(1)		Y if this materialized view includes a GROUP BY clause or aggregation, else N
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
INCREFRESHTIM	NUMBER		Approximate refresh time, in seconds (defined only when summary = Y for full refresh
CONTAINS_VIEWS	VARCHAR2(1)		Y if this materialized view contains a view in its definition, else N
UNUSABLE	VARCHAR2(1)		Y if this materialized view is in an unusable state (inconsistent data), else N. A materialized view can be in an unusable state if a system failure occurs during a full refresh
RESTRICTED_SYNTAX	VARCHAR2(1)		Y if this materialized view uses restricted materialized view syntax, else N
INC_REFRESHABLE	VARCHAR2(1)		Y if this materialized view can be incrementally refreshed, else N
KNOWN_STALE	VARCHAR2(1)		Y if the data contained in the materialized view is known to be inconsistent with the detail data, else N
INVALID	VARCHAR2(1)		Y if this materialized view is in an invalid state (inconsistent metadata), else N
QUERY_LEN	NUMBER	NOT NULL	The length of the query field
QUERY	LONG	NOT NULL	SELECT expression of the materialized view definition
REVISION	NUMBER		The materialized view metadata revision level

USER_MVIEW_DETAIL_RELATIONS

This view 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. Inline views in the materialized view definition are not represented in this table.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name, eg, the name of a table or view

Column	Datatype	Null	Description
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, MATERIALIZED VIEW, or SNAPSHOT
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

USER_MVIEW_JOINS

This view represents a join between two columns in the WHERE clause of a materialized view.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or TABLE
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

USER_MVIEW_KEYS

This view 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. Inline views in the materialized view definition are not represented in this table.

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	Position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	The 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, eg, the name of a table or view
DETAILOBJ_TYPE	VARCHAR2(5)		VIEW or TABLE
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

Column	Datatype	Null	Description
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

USER_NESTED_TABLES

This view describes the nested tables contained in the user's own tables.

Column	Datatype	NULL	Description
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)		Indication of default or user-specified storage for the varray
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column locator or value

USER_OBJECT_TABLES

This view contains descriptions of the object tables available to the user.

Column	Datatype	NULL	Description
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 of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes

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)		Has table been backed up since last modification?
NUM_ROWS	NUMBER		The number of rows in the table
BLOCKS	NUMBER		The number of used blocks in the table
EMPTY_BLOCKS	NUMBER		The number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		The average available free space in the table
CHAIN_CNT	NUMBER		The number of chained rows in the table
AVG_ROW_LEN	NUMBER		The average row length, including row overhead
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 table is to be cached in the buffer cache
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this table
LAST_ANALYZED	DATE		The date of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Is this table partitioned? YES or NO
IOT_TYPE	VARCHAR2(12)		If an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)	NOT NULL	Type of the table if the table is a typed table
PACKED	VARCHAR2(1)		If the table is a typed table, does it store objects in packed format?

Column	Datatype	NULL	Description
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		The 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)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

USER_OBJECTS

This view lists objects owned by the user.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(128)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the sub-object (for example, partition)
OBJECT_ID	NUMBER		Object number of the object
DATA_OBJECT_ID	NUMBER		Object number of the segment which contains the object
OBJECT_TYPE	VARCHAR2(15)		Type of the object (eg, TABLE, INDEX)
CREATED	DATE		Timestamp for the creation of the object
LAST_DDL_TIME	DATE		Timestamp of the last DDL command applied to the object (including grants and revokes)
TIMESTAMP	VARCHAR2(19)		Timestamp for the creation of the object (character data)
STATUS	VARCHAR2(7)		Status of the object: VALID, INVALID
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
GENERATED	VARCHAR2(1)		Was the name of this object system generated?
SECONDARY	VARCHAR2(1)		Is this a secondary object created as part of create for domain indexes?

USER_OBJECT_SIZE

This view lists the sizes, in bytes, of various PL/SQL objects.

Column	Datatype	NULL	Description
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 source code in bytes. Must be in memory during compilation, or dynamic recompilation
PARSED_SIZE	NUMBER		Size of parsed code in bytes. Must be in memory when an object is being compiled that references this object
CODE_SIZE	NUMBER		Size of compiled code 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

USER_OBJ_AUDIT_OPTS

This view, created by CATAUDIT.SQL, lists auditing options for all objects owned by a user.

Column	Datatype	NULL	Description
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

Column	Datatype	NULL	Description
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
CRE	VARCHAR2(3)		Auditing CREATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing READ WHENEVER SUCCESSFUL / UNSUCCESSFUL
WRI	VARCHAR2(3)		Auditing WRITE WHENEVER SUCCESSFUL / UNSUCCESSFUL

USER_OPANCILLARY

This view lists ancillary information for operators.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of primary operator

USER_OPARGUMENTS

This view lists argument information for operator bindings.

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
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

USER_OPBINDINGS

This view lists operator bindings.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		Name of the schema of the return type - not null only for ADTs
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_ TYPE_SCHEMA	VARCHAR2(30)		Schema of the implementation type of the indextype
IMPLEMENTATION_ TYPE	VARCHAR2(30)		Implementation type of the indextype

USER_OPERATORS

This view lists operators.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

USER_OUTLINE_HINTS

This view lists the set of hints which make up the outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline
NODE	NUMBER		I.D. of the query or subquery to which the hint applies. The top-level query is labelled 1. Subqueries are assigned sequentially numbered labels, starting with 2
JOIN_POS	NUMBER		Position of the table in the join order. The JOIN_POS column is 0 for all hints except the access method hints. The access method hints identify a table to which the hint and the join position apply
HINT	VARCHAR2(512)		Text of the hint

USER_OUTLINES

This view lists information about outlines.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		The user-specified or generated name of the stored outline. The name must be of a form that can be expressed in SQL
CATEGORY	VARCHAR2(30)		A user-defined name used to group outlines into collections
USED	VARCHAR2(9)		Flag indicating whether the outline has ever been used
TIMESTAMP	DATE		Timestamp at which the outline was created
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

USER_PART_COL_STATISTICS

This view contains column statistics and histogram information for table partitions that the current user owns.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2(30)		Table partition 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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

Column	Datatype	NULL	Description
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

USER_PART_HISTOGRAMS

This view contains the histogram data (end-points per histogram) for histograms on table partitions that the current user can access.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
PARTITION_NAME	VARCHAR2(30)		Table partition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

USER_PART_KEY_COLUMNS

This view describes the partitioning key columns for partitioned objects that the current user owns.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Partitioned table or index name
OBJECT_TYPE	VARCHAR2(11)		The object type ('TABLE' or 'INDEX')
COLUMN_NAME	VARCHAR2(30)		Column name
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

USER_PART_INDEXES

This view lists the object level partitioning information for all partitioned indexes that the user owns.

Column	Datatype	NULL	Description
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)		'In addition to RANGE, may specify HASH

Column	Datatype	NULL	Description
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this index
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	Number of partitions in this index
PARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY _COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
LOCALITY	VARCHAR2(6)		Indicates whether this partitioned index is LOCAL or GLOBAL
ALIGNMENT	VARCHAR2(12)		Indicates whether this partitioned index is PREFIXED or NON-PREFIXED
DEF_TABLESPACE _NAME	VARCHAR2(30)		Default TABLESPACE, for LOCAL index, for ADD / SPLIT TABLE partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, for LOCAL index, for ADD TABLE partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, for LOCAL index, for ADD TABLE partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, for LOCAL index, for ADD TABLE partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENTS, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE, for LOCAL index, for ADD TABLE partition, 'DEFAULT' if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, for LOCAL index, for ADD TABLE partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELISTS, for LOCAL index, for ADD TABLE partition

Column	Datatype	NULL	Description
DEF_LOGGING	VARCHAR2(7)		Default LOGGING, for LOCAL index, for ADD TABLE PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool, for LOCAL index, for ADD TABLE PARTITION

USER_PART_LOBS

This view describes table-level information for partitioned LOBs, including default attributes for LOB data partitions.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned table containing LOB(s)
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of a LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of a partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default CHUNK for a LOB data partition, used for ADD PARTITION
DEF_PCTVERSION	NUMBER	NOT NULL	Default PCTVERSION for a LOB data partition, used for ADD PARTITION
DEF_CACHE	VARCHAR2(3)		Default CACHE for a LOB data partition, used for ADD PARTITION
DEF_IN_ROW	VARCHAR2(3)		Default "IN ROW" for a LOB data partition, used for ADD PARTITION
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default TABLESPACE for a LOB data partition, used for ADD PARTITION
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL for a LOB data partition, used for ADD PARTITION
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT for a LOB data partition, used for ADD PARTITION
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENT for a LOB data partition, used for ADD PARTITION
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS for a LOB data partition, used for ADD PARTITION
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE for a LOB data partition, used for ADD PARTITION
DEF_FREELISTS	VARCHAR2(40)		Default FREELISTS for a LOB data partition, used for ADD PARTITION

Column	Datatype	NULL	Description
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default FREELIST GROUPS for a LOB data partition, used for ADD PARTITION
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition, used for ADD PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		Default BUFFER POOL for a LOB data partition, used for ADD PARTITION

USER_PART_TABLES

This view describes the object level partitioning information for partitioned tables that the current user owns.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		In addition to RANGE, may specify HASH
SUBPARTITIONING_TYPE	VARCHAR2(7)		May specify HASH or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this table
DEF_SUBPARTITION _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method, the default number of subpartitions, if any
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY _COUNT	NUMBER	NOT NULL	If the index is partitioned using a Composite method with Hash subpartitioning, this column will contain a number of columns in the subpartitioning key
DEF_TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Default TABLESPACE, used for ADD partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default PCTFREE, used for ADD partition
DEF_PCT_USED	NUMBER	NOT NULL	Default PCTUSED, used for ADD partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default INITRANS, used for ADD partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default MAXTRANS, used for ADD partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default INITIAL, used for ADD partition, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_NEXT_EXTENT	VARCHAR2(40)		Default NEXT, used for ADD partition, ’DEFAULT’ if attribute was not specified (in Oracle blocks)
DEF_MIN_EXTENTS	VARCHAR2(40)		Default MINEXTENTS, used for ADD partition, ’DEFAULT’ if attribute was not specified

Column	Datatype	NULL	Description
DEF_MAX_EXTENTS	VARCHAR2(40)		Default MAXEXTENTS, used for ADD partition, ’DEFAULT’ if attribute was not specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default PCTINCREASE, used for ADD partition, ’DEFAULT’ if attribute was not specified
DEF_FREELISTS	NUMBER	NOT NULL	Default FREELISTS, used for ADD partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default FREELIST GROUPS, used for ADD partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute, used for ADD PARTITION
DEF_BUFFER_POOL	VARCHAR2(7)		The default buffer pool for the given object

USER_PARTIAL_DROP_TABS

This view describes tables which have partially dropped tables.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

USER_PASSWORD_LIMITS

This view 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_POLICIES

This view lists policies on objects that are under the user’s schema.

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

Column	Datatype	NULL	Description
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?

USER_QUEUE_SCHEDULES

This view lists information about queue schedules.

Column	Datatype	NULL	Description
QNAME	VARCHAR2(30)	NOT NULL	Source queue name
DESTINATION	VARCHAR2(128)	NOT NULL	Destination name, currently limited to be a DBLINK name
START_DATE	DATE		Date to start propagation in the default date format
START_TIME	VARCHAR2(8)		Time of day at which to start propagation in HH:MI:SS format
PROPAGATION _WINDOW	NUMBER		Duration in seconds for the propagation window
NEXT_TIME	VARCHAR2(200)		Function to compute the start of the next propagation window
LATENCY	NUMBER		Maximum wait time to propagate a message during the propagation window
SCHEDULE_DISABLED	VARCHAR2(1)		N if enabled; Y if disabled and schedule will not be executed
PROCESS_NAME	VARCHAR2(8)		The name of the SNP background process executing this 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 OPS instance number executing this 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

Column	Datatype	NULL	Description
NEXT_RUN_DATE	DATE		The date at which the next window of this schedule will be started
NEXT_RUN_TIME	VARCHAR2(8)		The time of day at which the next window of this schedule will be started in HH:MI:SS format
TOTAL_TIME	NUMBER		The total time, in seconds, spent by the system in executing this schedule
TOTAL_NUMBER	NUMBER		The total number of messages propagated in this schedule
TOTAL_BYTES	NUMBER		The total number of bytes propagated in this schedule
MAX_NUMBER	NUMBER		The maximum number of messages propagated in a propagation window
MAX_BYTES	NUMBER		The maximum number of bytes propagated in a propagation window
AVG_NUMBER	NUMBER		The average number of messages propagated in a propagation window
AVG_SIZE	NUMBER		The average size of a propagated message in bytes
AVG_TIME	NUMBER		The average time, in seconds, to propagate a message
FAILURES	NUMBER		The number of times the execution failed. If 16, the schedule will be disabled
LAST_ERROR_DATE	DATE		The date of the last unsuccessful execution
LAST_ERROR_TIME	VARCHAR2(8)		The time of the last unsuccessful execution
LAST_ERROR_MSG	VARCHAR2(4000)		The error number and error message text of the last unsuccessful execution

USER_QUEUE_TABLES

This view describes only the queues in the queue tables created in the user's schema. For more information about this view and Advanced Queuing, see the *Oracle8i Application Developer's Guide - Fundamentals*.

Column	Datatype	NULL	Description
QUEUE_TABLE	VARCHAR2(30)		Name of the queue table
TYPE	VARCHAR2(7)		Type of payload: 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

Column	Datatype	NULL	Description
RECIPIENTS	VARCHAR2(8)		SINGLE or MULTIPLE recipients
MESSAGE_GROUPING	VARCHAR2(13)		NONE or TRANSACTIONAL
COMPATIBLE	VARCHAR2(5)		The lowest release level which this is compatible with (eg, 8.0.4)
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		The instance number of the instance which currently owns the queue table
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user

USER_QUEUES

This view describes the operational characteristics for every queue in the user's schema. For more information about this view and Advanced Queuing, see the *Oracle8i Application Developer's Guide - Fundamentals.*

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	The name of the queue table where this queue resides
QID	NUMBER	NOT NULL	Unique queue identifier
QUEUE_TYPE	VARCHAR2(20)		Queue type: NORMAL_QUEUE - Normal queue EXCEPTION_QUEUE - Exception queue NON_PERSISTENT_QUEUE - Non-persistent queue
MAX_RETRIES	NUMBER		Number of dequeue attempts allowed
RETRY_DELAY	NUMBER		Time lapse in seconds before retry takes place
ENQUEUE_ENABLED	VARCHAR2(7)		YES / NO
DEQUEUE_ENABLED	VARCHAR2(7)		YES / NO
RETENTION	VARCHAR2(40)		Number of seconds message is retained after dequeue. FOREVER - messages stay in the queue permanently
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user

USER_REFRESH

This view lists all the refresh groups.

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 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)		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 or 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
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

USER_REFRESH_CHILDREN

This view lists all the objects in refresh groups, where the user owns the refresh group.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is removed
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N, if Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		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 or 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
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

USER_REFS

This view describes the REF columns and REF attributes in the object type columns of the user's tables.

Column	Datatype	NULL	Description
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)		Is the REF value stored with ROWID? (YES or NO)
IS_SCOPED	VARCHAR2(3)		Is the REF column scoped? (YES or NO)
SCOPE_TABLE_OWNER	VARCHAR2(30)		Name of the owner of the scope table, if it exists and is accessible by the user

Column	Datatype	NULL	Description
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

USER_REGISTERED_SNAPSHOTS

This view lists users' registered snapshots.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot
SNAPSHOT_SITE	VARCHAR2(128)	NOT NULL	Global name of the snapshot site.
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		Specifies whether the snapshot is updatable. YES if it is, NO if it is not. If set to NO, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Whether the snapshot uses rowids or primary key for fast refresh
SNAPSHOT_ID	NUMBER(38)		Identifier for the snapshot used by the master for fast refresh
VERSION	VARCHAR2(17)		Version of snapshot
QUERY_TXT	LONG		Original query of which this snapshot is an instantiation

USER_REPCATLOG

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPCOLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPCOLUMN_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPCONFLICT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPDDL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPGENERATED

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPGENOBJECTS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPGROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPGROUPED_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPKEY_COLUMNS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPOBJECT

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPPARAMETER_COLUMN

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPPRIORITY

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPPRIORITY_GROUP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPPROP

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPRESOLUTION

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPRESOL_STATS_CONTROL

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPRESOLUTION_METHOD

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPRESOLUTION_STATISTICS

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_REPSITES

This view is used with Advanced Replication. For more information, see *Oracle8i Replication*.

USER_RESOURCE_LIMITS

This view 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_ROLE_PRIVS

This view 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: Y / N (occurs if configuration parameter OS_ROLES = TRUE)

USER_RSRC_CONSUMER_GROUP_PRIVS

This view lists all resource consumer groups granted to the user.

Column	Datatype	NULL	Description
GRANTED_GROUP	VARCHAR2(30)		Granted consumer group name
GRANT_OPTION	VARCHAR2(3)		Whether grant was with the GRANT option
INITIAL_GROUP	VARCHAR2(3)		Whether consumer group is designated as the default

USER_RSRC_MANAGER_SYSTEM_PRIVS

This view lists all the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package.

Column	Datatype	NULL	Description
PRIVILEGE	VARCHAR2(40)	NOT NULL	What type of privilege the user has
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option

USER_RULESETS

This view lists information about rulesets.

Column	Datatype	NULL	Description
RULESET_NAME	VARCHAR2(30)	NOT NULL	The name of the ruleset
RULESET_STORAGE_TABLE	VARCHAR2(61)		The table in which the ruleset is stored (will be schema.table)
BASE_TABLE	VARCHAR2(61)		The table on which the rules are defined (will be schema.table)

USER_SEGMENTS

This view lists information about storage allocation for database segments belonging to a user’s objects.

Column	Datatype	NULL	Description
SEGMENT_NAME	VARCHAR2(81)		Name of the segment, if any
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
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
BYTES	NUMBER		Size of the segment in bytes
BLOCKS	NUMBER		Size of the segment in Oracle blocks
EXTENTS	NUMBER		Number of extents allocated to the segment
INITIAL_EXTENT	NUMBER		Size of the initial extent in Oracle blocks
NEXT_EXTENT	NUMBER		Size of the next extent to be allocated in Oracle blocks

Column	Datatype	NULL	Description
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percent by which to increase the size of the next extent to be allocated
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object

USER_SEQUENCES

This view lists descriptions of the user's sequences.

Column	Datatype	NULL	Description
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 the sequence is incremented
CYCLE_FLAG	VARCHAR2(1)		Does sequence wraparound 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 actually used. This value is *not* continuously updated during database operation. It is intended for use after a warm start or import

USER_SNAPSHOTS

This view lists snapshots that the user can view.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the view used by users and applications for viewing the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	The table name

Column	Datatype	NULL	Description
MASTER_VIEW	VARCHAR2(30)		View of the master table, owned by the snapshot owner, used for refreshes. this is obsolete in Oracle8*i* and is NULL
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table of which this snapshot is a copy
MASTER_LINK	VARCHAR2(128)		Database link name to the master site
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		If YES, then snapshot is updatable snapshot. If NO, then is a read-only snapshot
LAST_REFRESH	DATE		Date and time at the master site of the last refresh
ERROR	NUMBER		The number of failed automatic refreshes since last successful refresh
TYPE	VARCHAR2(8)		Type of refresh for all automatic refreshes: COMPLETE, FAST, FORCE
NEXT	VARCHAR2(200)		Date function used to compute next refresh dates
START_WITH	DATE		Date function used to compute next refresh dates
REFRESH_GROUP	NUMBER		All snapshots in a given refresh group get refreshed in the same transaction
REFRESH_METHOD	VARCHAR2(11)		Values used to drive a fast refresh of the snapshot
FR_OPERATIONS	VARCHAR2(10)		Status of generated fast refresh operations: (REGENERATE, VALID)
CR_OPERATIONS	VARCHAR2(10)		Status of generated complete refresh operations: (REGENERATE, VALID)
UPDATE_TRIG	VARCHAR2(30)		Obsolete. It is NULL for Oracle8*i* snapshots.
UPDATE_LOG	VARCHAR2(30)		The table that logs changes made to an updatable snapshots
QUERY	LONG		Original query of which this snapshot is an instantiation
MASTER_ROLLBACK_SEG	VARCHAR2(30		Rollback segment used at the main site.
STATUS	VARCHAR2(7)		The status of the contents of the snapshot
REFRESH_MODE	VARCHAR2(8)		This indicates how and when the snapshot will be refreshed
PREBUILT	VARCHAR2(3)		If YES, this snapshot uses a prebuilt table as the base table

USER_SNAPSHOT_LOGS

This view lists all snapshot logs owned by the user.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot log
MASTER	VARCHAR2(30)	NOT NULL	Name of the master table for which the log records changes
LOG_TABLE	VARCHAR2(30)	NOT NULL	Log table that holds the ROWIDs and timestamps of rows that changed in the master table
LOG_TRIGGER	VARCHAR2(30)		Obsolete. It is NULL for Oracle8*i* snapshots.
ROWIDS	VARCHAR2(3)		If YES, the snapshot log records rowid information
PRIMARY_KEY	VARCHAR2(3)		If YES, the snapshot log records primary_key information
FILTER_COLUMNS	VARCHAR2(3)		If YES, the snapshot log records filter column information
CURRENT_SNAPSHOTS	DATE		Date and time when the snapshot of the master was last refreshed
SNAPSHOT_ID	NUMBER(38)		Unique snapshot identifier.

USER_SNAPSHOT_REFRESH_TIMES

This view lists snapshot refresh times.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

USER_SOURCE

This view contains text source of all stored objects belonging to the user.

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

Column	Datatype	NULL	Description
TYPE	VARCHAR2(12)		Type of object: PROCEDURE, PACKAGE, FUNCTION, TYPE, TYPE BODY, or PACKAGE BODY
LINE	NUMBER	NOT NULL	Line number of this line of source
TEXT	VARCHAR2(4000)		Text source of the stored object

USER_SUBPART_COL_STATISTICS

This view lists column statistics and histogram information for table subpartitions.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying subpartitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

USER_SUBPART_HISTOGRAMS

This view lists the actual histogram data (end-points per histogram) for histograms on table subpartitions.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name

Column	Datatype	NULL	Description
SUBPARTITION_NAME	VARCHAR2(30)		Table partition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

USER_SUBPART_KEY_COLUMNS

This view lists subpartitioning key columns for tables (and Local indexes on tables) partitioned using the Composite Range / Hash method.

Column	Datatype	NULL	Description
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

USER_SYNONYMS

This view lists the user’s private synonyms.

Column	Datatype	NULL	Description
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)		Database link referenced in a remote synonym

USER_SYS_PRIVS

This view lists system privileges granted to the user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name of the user, or PUBLIC
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege granted to the user
ADMIN_OPTION	VARCHAR2(3)		Signifies the privilege was granted with ADMIN option

USER_TAB_COL_STATISTICS

This view contains column statistics and histogram information which is in the USER_TAB_COLUMNS view. For more information, see "USER_TAB_COLUMNS" on page 2-249.

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
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 of the most recent time this column was analyzed
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

USER_TAB_COLUMNS

This view contains information about columns of user's tables, views, and clusters. To gather statistics for this view, use the SQL command ANALYZE.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table, view, or cluster name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_LENGTH	NUMBER	NOT NULL	Maximum length of the column in bytes
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_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype; NULL for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Does column allow 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 HIGH_VALUE	RAW(32)		
DENSITY	NUMBER		
NUM_NULLS	NUMBER		The number of NULLs in the column
NUM_BUCKETS	NUMBER		The number of buckets in histogram for the column
LAST_ANALYZED	DATE		The date of the most recent time this column was analyzed
SAMPLE_SIZE	NUMBER		The sample size used in analyzing this column
PACKED	VARCHAR2(1)		Does column store values in packed format?
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS, NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		The declared length for CHAR, VARCHAR, VARCHAR2, NCAR, and NVARCHAR2 datatypes; NULL for all other datatypes

Column	Datatype	NULL	Description
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

USER_TAB_COMMENTS

This view contains comments on the tables and views owned by the user.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
TABLE_TYPE	VARCHAR2(11)		Type of the object: TABLE (indicating the value for object tables and regular tables) or VIEW
COMMENTS	VARCHAR2(4000)		Comment on the object

USER_TAB_HISTOGRAMS

This view lists histograms on columns of user’s tables.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket

USER_TAB_PARTITIONS

This view describes, for each table partition, the partition level partitioning information, the storage parameters for the partition, and various partition statistics determined by ANALYZE that the current user owns.

Column	Datatype	NULL	Description
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

Column	Datatype	NULL	Description
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
PARTITION_NAME	VARCHAR2(30)		Partition name
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 of the most recent time this partition was analyzed
BUFFER_POOL	VARCHAR2(7)		Actual buffer pool of the given object
GLOBAL_STATS	VARCHAR2(3)		
USER_STATS	VARCHAR2(3)		

USER_TAB_PRIVS

This view contains information on grants on objects for which the user is the owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privileges was granted with ADMIN OPTION; otherwise NO

USER_TAB_PRIVS_MADE

This view contains information about all grants on objects owned by the user.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
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

USER_TAB_PRIVS_RECD

This view contains information about grants on objects for which the user is the grantee.

Column	Datatype	NULL	Description
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

USER_TAB_SUBPARTITIONS

This view describes, for each table subpartition, its name, name of the table and partition to which it belongs, and its storage attributes. Note that statistics will not be collected on a per-subpartition basis.

Column	Datatype	NULL	Description
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

Column	Datatype	NULL	Description
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 size of the sample
LAST_ANALYZED	DATE		When last analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

USER_TABLES

This view contains a description of the user's own relational tables. To gather statistics for this view, use the SQL command ANALYZE.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned 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
PCT_FREE	NUMBER		Minimum percentage of free space in a block; NULL for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; NULL for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; NULL for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; NULL for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes; NULL for partitioned tables

Column	Datatype	NULL	Description
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes; NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; NULL for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; NULL for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to 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 modification
NUM_ROWS	NUMBER		Number of rows returned by the ANALYZE command
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 that 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 table is to be cached in the buffer cache
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 of the most recent time this table was analyzed
PARTITIONED	VARCHAR2(3)		Indicates whether this table is partitioned. Set to ’YES’ if it is partitioned

Column	Datatype	NULL	Description
IOT_TYPE	VARCHAR2(12)		If this is an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW. If this is not an index organized table, IOT_TYPE is NULL
TEMPORARY	VARCHAR2(1)		Can the current session see only the data that it places in this object itself?
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool for the appropriate object; NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		Are the statistics calculated without merging underlying partitions?
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
DURATION	VARCHAR2(15)		If temporary table, then duration is sys$session or sys$transaction else NULL
SKIP_CORRUPT	VARCHAR2(8)		Whether skip corrupt blocks is enabled or disabled

USER_TABLESPACES

This view contains descriptions of accessible tablespaces.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
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	NOT NULL	Default maximum number of extents
PCT_INCREASE	NUMBER	NOT NULL	Default percent increase for extent size
MIN_EXTLEN	NUMBER		Minimum extent size for the tablespace
STATUS	VARCHAR2(9)		Tablespace status: ONLINE, OFFLINE, or INVALID (tablespace has been dropped)
CONTENTS	VARCHAR2(9)		Tablespace contents: "PERMANENT", or "TEMPORARY"
LOGGING	VARCHAR2(9)		Default logging attribute
EXTENT_ MANAGEMENT	VARCHAR2(10)		Extent management tracking: "DICTIONARY" or "LOCAL"
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for this tablespace

USER_TRIGGERS

This view contains descriptions of the user's triggers.

Column	Datatype	NULL	Description
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(75)		Events that fire the trigger: INSERT, UPDATE, DELETE, STARTUP, SHUTDOWN, ERROR, LOGON, LOGOFF, CREATE, ALTER, DROP
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
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)		WHEN clause. Must evaluate to TRUE for TRIGGER_BODY to execute.
STATUS	VARCHAR2(8)		Whether the trigger is enabled: ENABLED or 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

USER_TRIGGER_COLS

This view displays the usage of columns in triggers owned by the user or on one of the user's tables.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)	NOT NULL	Owner of the trigger

Column	Datatype	NULL	Description
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.

USER_TS_QUOTAS

This view contains information about tablespace quotas for the user.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
BYTES	NUMBER		Number of bytes charged to the user
MAX_BYTES	NUMBER		User’s quota in bytes, or -1 for UNLIMITED
BLOCKS	NUMBER	NOT NULL	Number of Oracle blocks charged to the user
MAX_BLOCKS	NUMBER		User’s quota in Oracle blocks, or -1 or UNLIMITED

USER_TYPES

This view displays the user’s types in a table.

Column	Datatype	NULL	Description
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
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type

USER_TYPE_ATTRS

This view displays the attributes of the user's types.

Column	Datatype	NULL	Description
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
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)		The name of the character set: CHAR_CS, 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 ID number)

USER_TYPE_METHODS

This view is a description of the user's methods types.

Column	Datatype	NULL	Description
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER		Method number for distinguishing overloaded method (not to be used as ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method

USER_UNUSED_COL_TABS

This view contains a description of all tables containing unused columns.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	The name of the table
COUNT	NUMBER		The number of unused columns

USER_UPDATABLE_COLUMNS

This view contains a description of columns that are updatable to the user in a join view.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

USER_USERS

This view contains information about the current user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
ACCOUNT_STATUS	VARCHAR2(32)	NOT NULL	Indicates if the account is locked, expired, or unlocked
LOCK_DATE	DATE		Date the account was locked if account status is locked
EXPIRY_DATE	DATE		Date of expiration of the account if account status is expired
DEFAULT_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for data
TEMPORARY_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for temporary tables
CREATED	DATE	NOT NULL	User creation date

Column	Datatype	NULL	Description
INITIAL_RSRC_CONSUMER_GROUP	VARCHAR2(30)		The initial resource consumer group for the user
EXTERNAL_NAME	VARCHAR2(4000)		User external name

USER_USTATS

This view contains information about 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
PROPERTY	VARCHAR2(6)		Property of the object - column or index
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

USER_VARRAYS

This view lists the text of views accessible to the user.

Column	Datatype	NULL	Description
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

USER_VIEWS

This view contains the text of views owned by the user.

Column	Datatype	NULL	Description
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

3

Dynamic Performance (V$) Views

This chapter describes the dynamic performance views These views are typically referred to as V$ views.

The following topics are included in this chapter:

- Dynamic Performance Views
- View Descriptions

Dynamic Performance Views

The Oracle server 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. These views can be selected from, but never updated or altered by the user.

The file CATALOG.SQL contains definitions of the views and public synonyms for the dynamic performance views. You must run CATALOG.SQL to create these views and synonyms.

V$ Views

Dynamic performance views are identified by the prefix V_$. Public synonyms for these views have the prefix V$. Database administrators or users should only access 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.

Suggestion: Once the instance is started, the V$ views that read from memory are accessible. Views that read data from disk require that the database be mounted.

Warning: Information about the dynamic performance views is presented for completeness only; this information does not imply a commitment to support these views in the future.

GV$ Views

In Oracle, there is an additional class of fixed views, the GV$ (Global V$) fixed views. For each of the V$ views described in this chapter (with the exception of V$CACHE_LOCK, V$LOCK_ACTIVITY, V$LOCKS_WITH_COLLISIONS, and V$ROLLNAME), there is a GV$ view. In a parallel server environment, querying a GV$ view retrieves the V$ view information from all qualified instances. In addition to the V$ information, each GV$ view possesses an additional column named INST_ID with type integer. 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 query:

```
SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5
```

retrieves the information from the V$ views on instances 2 and 5.

The GV$ views can be used to return information on groups of instances defined with the OPS_ADMIN_GROUP parameter. For more information see "OPS_ADMIN_GROUP" and *Oracle8i Parallel Server Concepts and Administration*.

Restrictions on GV$ Views

GV$ views have the following restrictions:

- The value of the PARALLEL_MAX_SERVERS parameter must be greater than zero on all instances mounting the database.
- The OPS_ADMIN_GROUP parameter must be defined with at least one member for a query to successfully complete.

Access to the Dynamic Performance Tables

After installation, only username SYS or anyone with SYSDBA ROLE has access to the dynamic performance tables.

For more information, see *Oracle Enterprise Manager Administrator's Guide*.

View Descriptions

This section lists 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(12)	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$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'
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. For more information, see "V$LOG" on page 3-62.

Column	Datatype	Description
GROUP#	NUMBER	Log file group number
THREAD#	NUMBER	Log file thread number
SEQUENCE#	NUMBER	Log file sequence number
CURRENT	VARCHAR2(3)	Archive log currently in use
FIRST_CHANGE#	NUMBER	First SCN stored in the current log

V$ARCHIVE_DEST

This view describes, for the current instance, all the archive log destinations, their current value, mode, and status.

Column	Datatype	Description
DEST_ID	NUMBER	ID (1-5)
STATUS	VARCHAR2(9)	Status: 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
BINDING	VARCHAR2(9)	Requirement for success: MANDATORY- must succeed or OPTIONAL - need not succeed (depends on LOG_ARCHIVE_MIS_SUCCEED_DEST)
NAME_SPACE	VARCHAR2(7)	Definition scope: SYSTEM-System definition or SESSION-Session definition
TARGET	VARCHAR2(7)	Target: PRIMARY-Copy to primary or STANDBY-Copy to standby
REOPEN_SECS	NUMBER	Retry time in seconds (after error)
DESTINATION	VARCHAR2(256)	Destination text string (translated primary location or standby service name)
FAIL_DATE	DATE	Date and time of any last error
FAIL_SEQUENCE	NUMBER	Any log sequence number at last error
FAIL_BLOCK	NUMBER	Any block number at last error
ERROR	VARCHAR2(256)	Text of any last error

For more information on archived log destinations, see "LOG_ARCHIVE_DEST" on page 1-60, "LOG_ARCHIVE_DUPLEX_DEST" on page 1-63, and "LOG_ARCHIVE_DEST_n" on page 1-61 "LOG_ARCHIVE_DEST_STATE_n" on page 1-62, "STANDBY_ARCHIVE_DEST" on page 1-121, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-65.

V$ARCHIVED_LOG

This view displays archived log information from the controlfile 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.

Column	Datatype	Description
RECID	NUMBER	Archived log record ID
STAMP	NUMBER	Archived log record stamp
NAME	VARCHAR2(512)	Archived log file name. If set to NULL, the log file was cleared before it was archived
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
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
ARCHIVED	VARCHAR2(3)	YES/NO
DELETED	VARCHAR2(3)	YES/NO
COMPLETION_TIME	DATE	Time when the archiving completed

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$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 backup set information from the controlfile. A backup set record is inserted after the backup set is successfully completed.

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

Column	Datatype	Description
LONG_WAIT_TIME _MAX	NUMBER	The maximum time taken for a blocking wait for I/O completion, in 100ths of seconds

V$BACKUP_CORRUPTION

This view displays information about corruptions in datafile backups from the controlfile. Note that corruptions are not tolerated in the controlfile and archived 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 number
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 backup

V$BACKUP_DATAFILE

This view displays backup datafile and backup controlfile information from the controlfile.

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

Column	Datatype	Description
FILE#	NUMBER	Datafile number. Set to 0 for controlfile
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 controlfile. 0 for datafile backups
COMPLETION_TIME	DATE	The time completed

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 controlfile. 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 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
TAG	VARCHAR2(32)	Backup piece tag. The tag is specified at backup set level, but stored at piece level
DELETED	VARCHAR2(3)	If set to YES indicates the piece is deleted, otherwise set to NO
START_TIME	DATE	The starting time

Column	Datatype	Description
COMPLETION_ TIME	DATE	The completion time
ELAPSED_ SECONDS	NUMBER	The number of elapsed seconds

V$BACKUP_REDOLOG

This view displays information about archived logs in backup sets from the controlfile. 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 log was switched out of. The redo in the log is below this SCN
NEXT_TIME	DATE	Time allocated when the log was switched out of
BLOCKS	NUMBER	Size of the log in logical blocks including the header block
BLOCK_SIZE	NUMBER	Size of the log blocks in bytes

V$BACKUP_SET

This view displays backup set information from the controlfile. 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 timestamp
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$BACKUP_CORRUPTION
SET_COUNT	NUMBER	Backup set count. The backup set count is incremented by one every time a new backup set is started (if the backup set is never completed the number is "lost"). If the controlfile is recreated then the count is reset to 1. Therefore the count must be used with the stamp to uniquely identify a backup set Primary key for the V$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 controlfile 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, non-zero 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	When the backup completes successfully, this is set to the completion time. This is the same time that was returned by backupEnd. If the backup is still in progress or has failed, this is set to NULL
ELAPSED_ SECONDS	NUMBER	The number of elapsed seconds
BLOCK_SIZE	NUMBER	Block size of the backup set

V$BACKUP_SYNC_IO

This view displays backup set information from the controlfile. A backup set record is inserted after the backup set is successfully completed.

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

Column	Datatype	Description
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	Name of this background process
DESCRIPTION	VARCHAR2	Description of the background process
ERROR	NUMBER	Error encountered

V$BH

This is a Parallel Server 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)	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.

Column	Datatype	Description
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.
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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

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. For more information, see "DB_BLOCK_LRU_LATCHES" on page 1-27.

Column	Datatype	Description
ID	NUMBER	Buffer pool ID number
NAME	VARCHAR2	Buffer pool name
LO_SETID	NUMBER	Low set ID number
HI_SETID	NUMBER	High set ID number

Column	Datatype	Description
SET_COUNT	NUMBER	Number of sets in this buffer pool. This is HI_SETID - LO_SETID + 1
BUFFERS	NUMBER	Number of buffers allocated to the buffer pool
LO_BNUM	NUMBER	Low buffer number for this pool
HI_BNUM	NUMBER	High buffer number for this pool

V$BUFFER_POOL_STATISTICS

This view displays information about all buffer pools available for the instance. The "sets" pertain to the number of LRU latch sets. For more information, see "DB_BLOCK_LRU_LATCHES" on page 1-27.

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
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

Column	Datatype	Description
PHYSICAL_WRITES	NUMBER	Physical writes statistic

V$CACHE

This is a Parallel Server view. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find 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
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. See Table 3–1
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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

Table 3–1 Values for the KIND column

Type Number	KIND Value	Type Number	KIND Value
1	INDEX	11	PACKAGE BODY
2	TABLE	12	TRIGGER
3	CLUSTER	13	TYPE
4	VIEW	14	TYPE BODY
5	SYNONYM	19	TABLE PARTITION
6	SEQUENCE	20	INDEX PARTITION
7	PROCEDURE	21	LOB
8	FUNCTION	22	LIBRARY
9	PACKAGE	NULL	UNKNOWN
10	NON-EXISTENT	------	-------

V$CACHE_LOCK

This is a Parallel Server view.

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. See Table 3–1

Column	Datatype	Description
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
INDX	NUMBER	Platform-specific lock manager identifier
CLASS	NUMBER	Platform-specific lock manager identifier

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$BH to find the buffers that are covered by the lock. See also "V$CACHE" on page 3-18.

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

V$CIRCUIT

This view 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

Column	Datatype	Description
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
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_PING

V$CLASS_PING displays the number of blocks pinged per block class. Use this view to compare contentions for blocks in different classes.

Column	Datatype	Description
CLASS	NUMBER	Number that represents 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
X_2_SSX	NUMBER	Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks of the specified CLASS

Column	Datatype	Description
X_2_SSX_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-Sub Shared Exclusive 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
SS_2_NULL	NUMBER	Number of lock conversions from Sub Shared-to-NULL for all blocks of the specified CLASS
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
SSX_2_X	NUMBER	Number of lock conversions from Sub Shared Exclusive-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
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub 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 name of namespace
ATTRIBUTE	VARCHAR2(30)	The name of attribute
VALUE	VARCHAR2(64)	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 controlfile record sections.

Column	Datatype	Description
TYPE	VARCHAR2(17)	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
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 controlfile.

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

V$DATABASE

This view contains database information from the control file.

Column	Datatype	Description
DBID	NUMBER	The database ID
NAME	VARCHAR2	Name of the database
CREATED	DATE	Creation date
LOG_MODE	VARCHAR2	Archive log mode: NOARCHIVELOG or ARCHIVELOG
CHECKPOINT_ CHANGE#	NUMBER	Last SCN checkpointed
ARCHIVE_CHANGE#	NUMBER	Last SCN archived
DBID	NUMBER	Database ID calculated when database is created and stored in all file headers
RESETLOGS_CHANGE#	NUMBER	Change# at open resetlogs
RESETLOGS_TIME	DATE	Timestamp of open resetlogs
PRIOR_RESETLOGS_CHANGE#	NUMBER	Change# at prior resetlogs
PRIOR_RESETLOGS_TIME	DATE	Timestamp of prior resetlogs
CONTROLFILE_TYPE	VARCHAR2(7)	CURRENT / STANDBY / CLONE / BACKUP / CREATED. STANDBY indicates database is in standby mode. CLONE indicates a clone database. BACKUP / CREATED indicates database is being recovered using a backup or created controlfile. A standby database activate or database open after recovery changes the type to CURRENT
CONTROLFILE_CREATED	DATE	Controlfile creation timestamp
CONTROLFILE_SEQUENCE#	NUMBER	Controlfile sequence number incremented by controlfile transactions
CONTROLFILE_CHANGE#	NUMBER	Last change# in backup controlfile. Set to NULL if the controlfile is not a backup
CONTROLFILE_TIME	DATE	Last timestamp in backup controlfile. Set to NULL if the controlfile is not a backup
OPEN_RESETLOGS	VARCHAR2(11)	NOT ALLOWED / ALLOWED / REQUIRED. Indicates whether next database open allows or requires the resetlogs option
VERSION_TIME	DATE	The version time
OPEN_MODE	VARCHAR2(10)	Open mode information

V$DATAFILE

This view contains datafile information from the control file. See also the "V$DATAFILE_HEADER" on page 3-29 view which displays information from datafile headers.

Column	Datatype	Description
FILE#	NUMBER	File identification number
STATUS	VARCHAR2	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. It is one of the values in Table 3–1
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
BYTES	NUMBER	Current size in bytes; 0 if inaccessible
CREATE_BYTES	NUMBER	Size when created, in bytes
NAME	VARCHAR2	Name of the file
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
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
BLOCKS	NUMBER	Current datafile size in blocks; 0 if inaccessible

Column	Datatype	Description
BLOCK_SIZE	NUMBER	Block size of the datafile
NAME	VARCHAR2(512)	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.

Table 3–2 describes values that can be entered in the ENABLED column.

Table 3–2 Values for the ENABLED Column

ENABLED Column Value	Description
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

V$DATAFILE_COPY

This view displays datafile copy information from the controlfile.

Column	Datatype	Description
RECID	NUMBER	Datafile copy record ID
STAMP	NUMBER	Datafile copy record stamp
NAME	VARCHAR2(512)	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

Column	Datatype	Description
CHECKPOINT_ CHANGE#	NUMBER	Checkpoint change# of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY _CHANGE#	NUMBER	Highest change seen when the datafile was copied
RECOVERY_FUZZY _CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY _TIME	DATE	Timestamp of the highest change written to the file by media recovery
ONLINE_FUZZY	VARCHAR2(3)	YES/NO. If set to YES, this is a copy taken using an operating system utility after a crash or offline immediate (or an invalid copy taken while datafile was online and the database open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent
BACKUP_FUZZY	VARCHAR2(3)	YES/NO. If set to YES, this is a copy taken using the BEGIN BACKUP/END BACKUP technique. Recovery will need to apply all redo up to the end backup marker to make this copy consistent
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt by this copy operation. That is, blocks that were not marked corrupted in the source datafile, but were detected and marked as corrupted during the copy operation
MEDIA_CORRUPT	NUMBER	Total number of media corrupt blocks. For example, blocks with checksum errors are marked media corrupt
LOGICALLY_CORRUPT	NUMBER	Total number of logically corrupt blocks. For example, applying redo for unrecoverable operations will mark affected blocks logically corrupt
BLOCKS	NUMBER	Size of the datafile copy in blocks (also the size of the datafile when the copy was made)
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE _RANGE	NUMBER	The RECID of the oldest offline range record in this controlfile copy. 0 for datafile copies
COMPLETION_TIME	DATE	Time when the copy was completed
DELETED	VARCHAR2(3)	YES/NO. If set to YES the datafile copy has been deleted or overwritten

V$DATAFILE_HEADER

This view displays datafile information from the datafile headers.

Column	Datatype	Description
FILE#	NUMBER	Datafile number (from controlfile)
STATUS	VARCHAR2(7)	ONLINE / OFFLINE (from controlfile)
ERROR	VARCHAR2(18)	NULL if the datafile header read and validation were successful. If the read failed then the rest of the columns are NULL. If the validation failed then the rest of columns may display invalid data. If there is an error then usually the datafile must be restored from a backup before it can be recovered or used.
FORMAT	NUMBER	Indicates the format for the header block. The possible values are 6, 7, 8, or 0. 6 - indicates Oracle Version 6 7 - indicates Oracle Version 7 8 - indicates Oracle Version 8 0 - indicates the format could not be determined (for example, the header could not be read)
RECOVER	VARCHAR2(3)	File needs media recovery YES / NO
FUZZY	VARCHAR2(3)	File is fuzzy YES / NO
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
TABLESPACE_NAME	VARCHAR2(30)	Tablespace name
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change#
RESETLOGS_TIME	DATE	Resetlogs timestamp
CHECKPOINT_CHANGE#	NUMBER	Datafile checkpoint change#
CHECKPOINT_TIME	DATE	Datafile checkpoint timestamp
CHECKPOINT_COUNT	NUMBER	Datafile checkpoint count
BYTES	NUMBER	Current datafile size in bytes
BLOCKS	NUMBER	Current datafile size in blocks
NAME	VARCHAR2(512)	Datafile name

V$DBFILE

This view lists all datafiles making up the database. This view is retained for historical compatibility. Use of V$DATAFILE is recommended instead. For more information, see "V$DATAFILE" on page 3-26.

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$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

Column	Datatype	Description
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
EXECUTIONS	NUMBER	Not used. To see actual execution counts, see "V$SQLAREA".
LOCKS	NUMBER	Number of users currently locking this object
PINS	NUMBER	Number of users currently pinning this object
KEPT	VARCHAR2(3)	YES or NO, depending on whether this object has been "kept" (permanently pinned in memory) with the PL / SQL procedure DBMS_SHARED_POOL.KEEP

V$DB_PIPES

This view displays the pipes that are currently in this database.

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$DELETED_OBJECT

This view displays information about deleted archived logs, datafile copies and backup pieces from the controlfile. 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)	ARCHIVED LOG / DATAFILE COPY / BACKUP PIECE. Type of the deleted object
OBJECT_RECID	NUMBER	Record ID of the deleted object
OBJECT_STAMP	NUMBER	Record timestamp of the deleted object

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	Dispatcher status: WAIT (idle), SEND (sending a message connection), RECEIVE (receiving a message), CONNECT (establishing a connection), DISCONNECT (handling a disconnect request), BREAK (handling a break), OUTBOUND (establishing an outbound connection)
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 MTS_DISPATCHERS configuration used by this dispatcher

V$DISPATCHER_RATE

This view provides rate statistics for the dispatcher processes.

Column	Datatype	Description
NAME	CHAR	Process name
PADDR	RAW	Process address
CUR_LOOP_RATE	NUMBER	Current rate of loop events
CUR_EVENT_RATE	NUMBER	Current rate of events
CUR_EVENTS_PER_LOOP	NUMBER	Current events per loop
CUR_MSG_RATE	NUMBER	Current rate of messages
CUR_SVR_BUF_RATE	NUMBER	Current rate of buffers for the server
CUR_SVR_BYTE_RATE	NUMBER	Current rate of bytes for the server
CUR_SVR_BYTE_PER_BUF	NUMBER	Current bytes per buffer for the server
CUR_CLT_BUF_RATE	NUMBER	Current rate of buffers for the client
CUR_CLT_BYTE_RATE	NUMBER	Current rate of bytes for the client
CUR_CLT_BYTE_PER_BUF	NUMBER	Current bytes per buffer for the client
CUR_BUF_RATE	NUMBER	Current rate of buffers
CUR_BYTE_RATE	NUMBER	Current rate of bytes
CUR_BYTE_PER_BUF	NUMBER	Current bytes per buffer
CUR_IN_CONNECT_RATE	NUMBER	Current inbound connects
CUR_OUT_CONNECT_RATE	NUMBER	Current outbound connects
CUR_RECONNECT_RATE	NUMBER	Current reconnects for connection pool and multiplexing
MAX_LOOP_RATE	NUMBER	Maximum rate of loop events
MAX_EVENT_RATE	NUMBER	Maximum rate of events
MAX_EVENTS_PER_LOOP	NUMBER	Maximum events per loop
MAX_MSG_RATE	NUMBER	Maximum rate of messages
MAX_SVR_BUF_RATE	NUMBER	Maximum rate of buffers for the server
MAX_SVR_BYTE_RATE	NUMBER	Maximum rate of bytes for the server
MAX_SVR_BYTE_PER_BUF	NUMBER	Maximum number of bytes per buffer for the server
MAX_CLT_BUF_RATE	NUMBER	Maximum rate of buffers for the client

Column	Datatype	Description
MAX_CLT_BYTE_RATE	NUMBER	Maximum rate of bytes for the client
MAX_CLT_BYTE_PER_BUF	NUMBER	Maximum number of bytes per buffer for the client
MAX_BUF_RATE	NUMBER	Maximum rate of buffers
MAX_BYTE_RATE	NUMBER	Maximum rate of bytes
MAX_BYTE_PER_BUF	NUMBER	Maximum number of bytes per buffer
MAX_IN_CONNECT_RATE	NUMBER	Maximum number of inbound connects
MAX_OUT_CONNECT_RATE	NUMBER	Maximum number of outbound connects
MAX_RECONNECT_RATE	NUMBER	Maximum number of reconnects for connection pool and multiplexing
AVG_LOOP_RATE	NUMBER	Average rate of loop events
AVG_EVENT_RATE	NUMBER	Average rate of events
AVG_EVENTS_PER_LOOP	NUMBER	Average events per loop
AVG_MSG_RATE	NUMBER	Average rate of messages
AVG_SVR_BUF_RATE	NUMBER	Average rate of buffers for the server
AVG_SVR_BYTE_RATE	NUMBER	Average rate of bytes for the server
AVG_SVR_BYTE_PER_BUF	NUMBER	Average bytes per buffer for the server
AVG_CLT_BUF_RATE	NUMBER	Average rate of buffers for the client
AVG_CLT_BYTE_RATE	NUMBER	Average rate of bytes for the client
AVG_CLT_BYTE_PER_BUF	NUMBER	Average bytes per buffer for the client
AVG_BUF_RATE	NUMBER	Average rate of buffers
AVG_BYTE_RATE	NUMBER	Average rate of bytes
AVG_BYTE_PER_BUF	NUMBER	Average bytes per buffer
AVG_IN_CONNECT_RATE	NUMBER	Average inbound connects
AVG_OUT_CONNECT_RATE	NUMBER	Average outbound connects
AVG_RECONNECT_RATE	NUMBER	Average reconnects for connection pool and multiplexing
NUM_LOOPS_TRACKED	NUMBER	Number of loop tracked
NUM_MSG_TRACKED	NUMBER	Number of messages tracked
NUM_SVR_BUF_TRACKED	NUMBER	Number of buffers for the server tracked
NUM_CLT_BUF_TRACKED	NUMBER	Number of buffers for the client tracked

Column	Datatype	Description
NUM_BUF_TRACKED	NUMBER	Number of buffers tracked
NUM_IN_CONNECT _TRACKED	NUMBER	Number inbound connects tracked
NUM_OUT_CONNECT _TRACKED	NUMBER	Number outbound connects tracked
NUM_RECONNECT _TRACKED	NUMBER	Number of reconnects tracked
SCALE_LOOPS	NUMBER	Scale of loop
SCALE_MSG	NUMBER	Scale of messages
SCALE_SVR_BUF	NUMBER	Scale of buffers for the server
SCALE_CLT_BUF	NUMBER	Scale of buffers for the client
SCALE_BUF	NUMBER	Scale of buffers
SCALE_IN_CONNECT	NUMBER	Scale of inbound connects
SCALE_OUT_CONNECT	NUMBER	Scale of outbound connects
SCALE_RECONNECT	NUMBER	Scale of reconnects

V$DLM_ALL_LOCKS

This is a Parallel Server view. V$DLM_ALL_LOCKS lists information of all locks currently known to lock manager that are being blocked or blocking others.

Column	Datatype	Description
LOCKP	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

Column	Datatype	Description
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
LOCKSTATE	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$DLM_CONVERT_LOCAL

V$DLM_CONVERT_LOCAL displays the elapsed time for the local lock conversion operation.

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$DLM_CONVERT_REMOTE

V$DLM_CONVERT_REMOTE displays the elapsed time for the remote lock conversion operation.

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

Table 3–3 Values for the CONVERT_TYPE column

Conversion Type	Description
NULL -> SS	NULL mode to sub shared mode
NULL -> SX	NULL mode to shared exclusive mode
NULL -> S	NULL mode to shared mode
NULL -> SSX	NULL mode to sub-shared exclusive mode
NULL -> X	NULL mode to exclusive mode
SS -> SX	sub shared mode to shared exclusive mode
SS -> S	sub shared mode to shared mode

Table 3–3 Values for the CONVERT_TYPE column

Conversion Type	Description
SS -> SSX	sub shared mode to sub-shared exclusive mode
SS -> X	sub shared mode to exclusive mode
SX -> S	shared exclusive mode to shared mode
SX -> SSX	shared exclusive mode to sub-shared exclusive mode
SX -> X	shared exclusive mode to exclusive mode
S -> SX	shared mode to shared exclusive mode
S -> SSX	shared mode to sub-shared exclusive mode
S -> X	shared mode to exclusive mode
SSX -> X	sub-shared exclusive mode to exclusive mode

V$DLM_LATCH

V$DLM_LATCH is obsolete. See "V$LATCH" on page 3-52 for statistics about DLM latch performance.

V$DLM_LOCKS

This is a Parallel Server view. V$DLM_LOCKS lists information of all locks currently known to lock manager that are being blocked or blocking others.

Column	Datatype	Description
LOCKP	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

Column	Datatype	Description
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
LOCKSTATE	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$DLM_MISC

V$DLM_MISC displays miscellaneous DLM statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Name of the statistic
VALUE	NUMBER	Value associated with the statistic

V$DLM_RESS

V$DLM_RESS is a Parallel Server 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$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$LOCK. For more information, see "V$LOCK" on page 3-56.

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. For a list of user and system types that can have locks.
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)
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$EVENT_NAME

This view contains information about wait events.

Column	Datatype	Description
EVENT#	NUMBER	The number of the wait event

Column	Datatype	Description
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 a Parallel Server 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" on page 1-44 to reduce lock collisions.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_ FILES or V$DBFILES)
BLOCK#	NUMBER	Block number

Column	Datatype	Description
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. See Table 3–1
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

For more information, see "GC_FILES_TO_LOCKS" on page 1-44 and also *Oracle8i Parallel Server Concepts and Administration*.

V$FAST_START_SERVERS

V$FAST_START_SERVERS provides information about all the recovery slaves performing parallel transaction recovery.

Column	Datatype	Description
STATE	VARCHAR2(11)	State of the server; IDLE or RECOVERING
UNDOBLOCKSDONE	NUMBER	The percentage of the assigned work done so far
PID	NUMBER	The process ID

For further information, see *Oracle8i Backup and Recovery Guide*.

V$FAST_START_TRANSACTIONS

V$FAST_START_TRANSACTIONS contains information about the progress of the transactions that Oracle is recovering.

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

For further information, see *Oracle8i Backup and Recovery Guide*.

V$FILE_PING

The view V$FILE_PING 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
FREQUENCY	NUMBER	The frequency
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
X_2_SSX	NUMBER	Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks in the file
X_2_SSX_FORCED _WRITE	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
SS_2_RLS	NUMBER	Number of pcm locks sslocks released. 0 in Oracle 8.1
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 reuse block range cross instance calls for this file

Column	Datatype	Description
RBR_FORCED_STALE	NUMBER	Number of times a block in this file was made STALE due to reuse 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
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file
OP_2_SS	NUMBER	Number of pcm locks ss locks opened. 0 in Oracle 8.1

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
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	The average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE

Column	Datatype	Description
LSTIOTIM	NUMBER	The time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MINIOTIM	NUMBER	The 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	The maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIORTM	NUMBER	The 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$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
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 (listerner.ora SID)
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_SESSION

This view identifies the set of HS sessions currently open for the Oracle Server.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier
AGENT_ID	NUMBER	Foreign key to V$HS_AGENT
SID	NUMBER	User session identifier (foreign key to V$SESSION)
DB_LINK	VARCHAR2(128)	Server database link name used to access the agent NULL means that no database link is used (eg, when using external procedures)
DB_LINK_OWNER	NUMBER	Owner of the database link in DB_LINK
STARTTIME	DATE	Time the connection was initiated

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. This version of V$INSTANCE is not compatible with earlier versions of V$INSTANCE.

Column	Datatype	Description
INSTANCE_NUMBER	NUMBER	Instance number used for instance registration. Corresponds to INSTANCE_NUMBER initialization parameter See "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(7)	STARTED / MOUNTED / OPEN STARTED after startup nomount MOUNTED after startup mount or alter database close OPEN after startup or after database open
PARALLEL	VARCHAR2(3)	YES / NO in parallel server 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

Column	Datatype	Description
LOG_SWITCH_WAIT	VARCHAR2(11)	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

V$INSTANCE_RECOVERY

This view is used to monitor the mechanisms that implement the user-specifiable limit on recovery reads.

Column	Datatype	Description
RECOVERY_ EXTIMATED_IOS	NUMBER	The estimated number of blocks that would be processed during recovery, based on the in-memory value of the Fast-Start checkpoint.
ACTUAL_REDO_ BLOCKS	NUMBER	The current actual number of redo blocks required for recovery.
TARGET_REDO_ BLOCKS	NUMBER	The current target number of redo blocks that must be processed for recovery. It will be the minimum of the following four columns.
LOG_FILE_SIZE_ REDO_BLKS	NUMBER	Maximum number of redo blocks required to guarantee log switch does not occur before checkpoint completes.
LOG_CHKPT_ TIMEOUT_REDO_ BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy LOG_CHECKPOINT_TIMEOUT.
LOG_CHKPT_ INTERVAL_REDO_ BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy LOG_CHECKPOINT_INTERVAL.
FAST_START_IO_ TARGET_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy FAST_START_IO_TARGET.

V$LATCH

This view lists statistics for non-parent latches and summary statistics for parent latches. That is, the statistics for a parent latch include counts from each of its children.

Note: Columns SLEEP5, SLEEP6,... SLEEP11 are present for compatibility with previous versions of Oracle. No data are accumulated for these columns.

Column	Datatype	Description
ADDR	RAW(4)	Address of latch object
LATCH#	NUMBER	Latch number
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
GETS	NUMBER	Number of times obtained a wait
MISSES	NUMBER	Number of times obtained a wait but failed on the first try
SLEEPS	NUMBER	Number of times slept when wanted a wait
IMMEDIATE_GETS	NUMBER	Number of times obtained without a wait
IMMEDIATE_MISSES	NUMBER	Number of times failed to get without a wait
WAITERS_WOKEN	NUMBER	How many times a wait was awakened
WAITS_HOLDING _LATCH	NUMBER	Number of waits while holding a different latch
SPIN_GETS	NUMBER	Gets that missed first try but succeeded on spin
SLEEP1	NUMBER	Waits that slept 1 time
SLEEP2	NUMBER	Waits that slept 2 times
SLEEP3	NUMBER	Waits that slept 3 times
SLEEP4	NUMBER	Waits that slept 4 times
SLEEP5	NUMBER	Waits that slept 5 times
SLEEP6	NUMBER	Waits that slept 6 times
SLEEP7	NUMBER	Waits that slept 7 times
SLEEP8	NUMBER	Waits that slept 8 times
SLEEP9	NUMBER	Waits that slept 9 times
SLEEP10	NUMBER	Waits that slept 10 times
SLEEP11	NUMBER	Waits that slept 11 times

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$LATCH. For more information, see "V$LATCH" on page 3-52.

Column	Datatype	Description
LATCH#	NUMBER	Latch number
NAME	VARCHAR2(64)	Latch name

V$LATCH_CHILDREN

This view contains statistics about child latches. This view includes all columns of V$LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other. For more information, see "V$LATCH" on page 3-52.

Column	Datatype	Description
ADDR	RAW(4)	Address of latch object
LATCH#	NUMBER	Latch number for a parent latch
CHILD#	NUMBER	Child number of a parent latch shown in LATCH#
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
GETS	NUMBER	Number of times obtained a wait
MISSES	NUMBER	Number of times obtained a wait but failed on the first try
SLEEPS	NUMBER	Number of times slept when wanted a wait

Column	Datatype	Description
IMMEDIATE_GETS	NUMBER	Number of times obtained without a wait
IMMEDIATE_MISSES	NUMBER	Number of time failed to get without a wait
WAITERS_WOKEN	NUMBER	How many times a wait was awakened
WAITS_HOLDING_LATCH	NUMBER	Number of waits while holding a different latch
SPIN_GETS	NUMBER	Gets that missed first try but succeeded on spin
SLEEP*n*	NUMBER	Waits that slept *n* times

V$LATCH_MISSES

This view contains statistics about missed attempts to acquire a latch.

Column	Datatype	Description
PARENT_NAME	VARCHAR2	Latch name of a parent latch
WHERE	VARCHAR2	Location that attempted to acquire the latch
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	

V$LATCH_PARENT

This view contains statistics about the parent latch. The columns of V$LATCH_PARENT are identical to those in V$LATCH. For more information, see "V$LATCH" on page 3-52.

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

Column	Datatype	Description
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 meta data 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
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$LOADCSTAT

This view contains SQL*Loader statistics compiled during the execution of a direct load. These statistics apply to the whole load. Any SELECT against this table results in "no rows returned" since you cannot load data and do a query at the same time.

Column	Datatype	Description
READ	NUMBER	Number of records read
REJECTED	NUMBER	Number of records rejected
TDISCARD	NUMBER	Total number of discards during the load
NDISCARD	NUMBER	Number of discards from the current file

V$LOADTSTAT

SQL*Loader statistics compiled during the execution of a direct load. These statistics apply to the current table. Any SELECT against this table results in "no rows returned" since you cannot load data and do a query at the same time.

Column	Datatype	Description
LOADED	NUMBER	Number of records loaded
REJECTED	NUMBER	Number of records rejected
FAILWHEN	NUMBER	Number of records that failed to meet any WHEN clause
ALLNULL	NUMBER	Number of records that were completely null and were therefore not loaded
LEFT2SKIP	NUMBER	Number of records yet to skip during a continued load
PTNLOADED	NUMBER	Number of records loaded PTN

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

Column	Datatype	Description
TYPE	VARCHAR2(2)	Type of lock. For a list of user and system types that can have locks, see Table 3–4, "Values for the TYPE column: User Types" and Table 3–5, "Values for the TYPE column: System Types"
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)
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

The locks on the user types in Table 3–4 are obtained by user applications. Any process that is blocking others is likely to be holding one of these locks.

Table 3–4 Values for the TYPE column: User Types

User Type	Description
TM	DML enqueue
TX	Transaction enqueue
UL	User supplied

The locks on the system types in Table 3–5 are held for extremely short periods of time.

Table 3–5 Values for the TYPE column: System Types

System Type	Description
BL	Buffer hash table instance
CF	Control file schema global enqueue
CI	Cross-instance function invocation instance
CU	Cursor bind
DF	Data file instance
DL	Direct loader parallel index create
DM	Mount/startup db primary/secondary instance
DR	Distributed recovery process
DX	Distributed transaction entry
FS	File set
HW	Space management operations on a specific segment
IN	Instance number
IR	Instance recovery serialization global enqueue
IS	Instance state
IV	Library cache invalidation instance
JQ	Job queue
KK	Thread kick
LA .. LP	Library cache lock instance lock (A..P = namespace)
MM	Mount definition global enqueue
MR	Media recovery
NA..NZ	Library cache pin instance (A..Z = namespace)
PF	Password File
PI, PS	Parallel operation
PR	Process startup
QA..QZ	Row cache instance (A..Z = cache)
RT	Redo thread global enqueue
SC	System commit number instance

Table 3–5 Values for the TYPE column: System Types

System Type	Description
SM	SMON
SN	Sequence number instance
SQ	Sequence number enqueue
SS	Sort segment
ST	Space transaction enqueue
SV	Sequence number value
TA	Generic enqueue
TS	Temporary segment enqueue (ID2=0)
TS	New block allocation enqueue (ID2=1)
TT	Temporary table enqueue
UN	User name
US	Undo segment DDL
WL	Being-written redo log instance

V$LOCK_ACTIVITY

This is a Parallel Server view. V$LOCK_ACTIVITY displays the DLM lock operation activity of the current instance. Each row corresponds to a type of lock operation.

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

Column	Datatype	Description
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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

V$LOCK_ELEMENT

This is a Parallel Server view. There is one entry in v$LOCK_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}.

Column	Datatype	Description
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.
INDX	NUMBER	Platform specific lock manager identifier
CLASS	NUMBER	Platform specific lock manager identifier
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	Non-zero if PCM lock is being downgraded
ACQUIRING	NUMBER	Non-zero if PCM lock is being upgraded
INVALID	NUMBER	Non-zero if PCM lock is invalid. (A lock may become invalid after a system failure.)
FLAGS	NUMBER	Process level flags for the LE

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

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$LOCKS_WITH_COLLISIONS

This is a Parallel Server 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.

Column	Datatype	Description
LOCK_ELEMENT_ ADDR	RAW(4)	The 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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

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
STATUS	VARCHAR2(16)	Log status. The STATUS column can have the values in Table 3–6.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log

Table 3–6 describes values in the log STATUS column.

Table 3–6 Values for the STATUS Column

STATUS	Meaning
UNUSED	Indicates 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	Indicates this is the current redo log. This implies that the redo log is active. The redo log could be open or closed.
ACTIVE	Indicates 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	Indicates the log is being recreated as an empty log after an ALTER DATABASE CLEAR LOGFILE command. After the log is cleared, the status changes to UNUSED.
CLEARING _CURRENT	Indicates that 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	Indicates the log is no longer needed for instance recovery. It may be in use for media recovery. It might or might not be archived.

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. Use of V$LOG_HISTORY is recommended instead. For more information, see "V$LOG_HISTORY" on page 3-66.

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.

Column	Datatype	Description
SCN	NUMBER(15)	The system change number
TIMESTAMP	DATE	The timestamp
THREAD#	NUMBER	The thread number
LOG_ID	NUMBER	The log ID
XIDUSN	NUMBER	The transaction ID undo segment number
XIDSLOT	NUMBER	The transaction ID slot number
XIDSQN	NUMBER	The transaction ID log sequence number

Column	Datatype	Description
RBASQN	NUMBER	The RBA log sequence number
RBABLK	NUMBER	The RBA block number
RBABYTE	NUMBER	The RBA byte offset
UBAFIL	NUMBER	The UBA file number
UBABLK	NUMBER	The UBA block number
UBAREC	NUMBER	The UBA record index
UBASQN	NUMBER	The UBA undo block sequence number
ABS_FILE#	NUMBER	The data block absolute file number
REL_FILE#	NUMBER	The data block relative file number
DATA_BLK#	NUMBER	The data block number
DATA_OBJ#	NUMBER	The data block object number
DATA_DOBJ#	NUMBER	The data block data object number
SEG_OWNER	VARCHAR2(30)	The owner name of the segment
SEG_NAME	VARCHAR2(81)	The segment name
SEG_TYPE	NUMBER	The segment type
TABLE_SPACE_NAME	VARCHAR2(30)	The tablespace name of segment
ROW_ID	VARCHAR2(18)	The row ID
SESSION#	NUMBER	The session number
SERIAL#	NUMBER	The serial number
USER_NAME	VARCHAR2(30)	The user name
SESSION_INFO	VARCHAR2(4000)	Session information
ROLLBACK	NUMBER	The rollback request
OPERATION	VARCHAR2(30)	The operation
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

Column	Datatype	Description
STATUS	VARCHAR2 (16)	The status

V$LOGMNR_DICTIONARY

This view contains log history information.

Column	Datatype	Description
TIMESTAMP	DATE	The date the dictionary was created
DB_ID	NUMBER	The database ID
DB_NAME	VARCHAR2(8)	The name of the database
FILENAME	VARCHAR2(513)	The dictionary filename
DICTIONARY_SCN	NUMBER	The system change number when the dictionary was created
RESET_SCN	NUMBER	The reset log SCN when the dictionary was created
RESET_SCN_TIME	NUMBER	The 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. The value of this field is also reported in the LOG_ID column of the V$LOG
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

Column	Datatype	Description
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$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	Controlfile record ID
STAMP	NUMBER	Controlfile record stamp

V$MLS_PARAMETERS

This is a Trusted Oracle Server view that lists Trusted Oracle Server-specific initialization parameters. For more information, see your Trusted Oracle documentation.

V$MTS

This view contains information for tuning the multi-threaded server.

Column	Datatype	Description
MAXIMUM_CONNECTIONS	NUMBER	The maximum number of connections each dispatcher can support. This value is determined at startup time using Net8 constants and other port-specific information, or can be lowered using the MTS_DISPATCHERS parameter
SERVERS_STARTED	NUMBER	The total number of multi-threaded servers started since the instance started (but not including those started during startup)
SERVERS_TERMINATED	NUMBER	The total number of multi-threaded servers stopped by Oracle since the instance started
SERVERS_HIGHWATER	NUMBER	The highest number of servers running at one time since the instance started. If this value reaches the value set for the MTS_MAX_SERVERS initialization parameter, consider raising the value of MTS_SERVERS. For more information, see "MTS_SERVERS" on page 1-76

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. For more information, see "V$SESSION" on page 3-92 and "V$SQL" on page 3-108.

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$DB_OBJECT_CACHE
TO_HASH	NUMBER	The hash value of the object that is depended on. These can be used to look up more information on the object in V$DB_OBJECT_CACHE
TO_TYPE	NUMBER	The type of the object that is depended on

V$OBSOLETE_PARAMETER

This view lists obsolete parameters. If any value is true, you should examine why.

Column	Datatype	Description
NAME	VARCHAR2(64)	The name of the parameter
ISSPECIFIED	VARCHAR2(5)	Whether the parameter was specified in the config file

V$OFFLINE_RANGE

This view displays datafile offline information from the controlfile. Note that the last offline range of each datafile is kept in the DATAFILE record. For more information, see "V$DATAFILE" on page 3-26.

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.

Column	Datatype	Description
RECID	NUMBER	Record ID
STAMP	NUMBER	Record stamp
FILE#	NUMBER	Datafile number

Column	Datatype	Description
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

This view lists information about initialization parameters.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(64)	Parameter name
TYPE	NUMBER	Parameter type; 1 = Boolean; 2 = string; 3 = integer
VALUE	VARCHAR2(512)	Parameter value
ISDEFAULT	VARCHAR2(9)	Whether the parameter value is the default
ISSES_ MODIFIABLE	VARCHAR2(5)	TRUE = the parameter can be changed with ALTER SESSION FALSE= the parameter cannot be changed with ALTER SESSION
ISSYS_ MODIFIABLE	VARCHAR2(9)	IMMEDIATE = the parameter can be changed with ALTER SYSTEM DEFERRED=the parameter cannot be changed until the next session FALSE= the parameter cannot be changed with ALTER SYSTEM
ISMODIFIED	VARCHAR2(10)	Indicates how the parameter was modified. If an ALTER SESSION was performed, the value will be MODIFIED. If an ALTER SYSTEM (which will cause all the currently logged in sessions' values to be modified) was performed the value will be SYS_MODIFIED
ISADJUSTED	VARCHAR2(5)	Indicates that the rdbms adjusted the input value to a more suitable value (e.g., the parameter value should be prime, but the user input a non-prime number, so the rdbms adjusted the value to the next prime number)
DESCRIPTION	VARCHAR2(64)	A descriptive comment about the parameter

V$PING

This is a Parallel Server view. The V$PING view is identical to the V$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. For more information, see "V$CACHE" on page 3-18.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query "DBA_DATA_FILES" on page 2-85 or "V$DBFILE" on page 3-30)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number

Column	Datatype	Description
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
KIND	VARCHAR2(15)	Type of database object. See Table 3–1
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

For more information, see *Oracle8i Parallel Server Concepts and Administration*.

V$PQ_SESSTAT

This view lists session statistics for parallel queries.

Note: This view will be obsoleted in a future release.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic. See Table 3–7
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

The statistics (fixed rows) in Table 3–7 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.

Table 3–7 Names of Statistics in the STATISTIC Column

Statistic (Fixed Row)	Description
Queries Parallelized	Number of queries that were run in parallel
DML Parallelized	Number of DML operations that were run in parallel
DFO Trees	Number of executed DFO trees
Server Threads	Total number of parallel servers 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

V$PQ_SLAVE

This view lists statistics for each of the active parallel execution servers on an instance.

Note: This view will be replaced / obsoleted in a future release by a new view called V$PX_PROCESS.

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 or 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.

Note: This view will be replaced / obsoleted in a future release by a new view called V$PX_PROCESS_SYSSTAT.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic. See Table 3–8
VALUE	NUMBER	The value of the statistic

The statistics (fixed rows) in Table 3–8 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.

Table 3–8 Names of Statistics in the STATISTIC Column

Statistic (Fixed Row)	Description
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

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.

Note: This view will be renamed V$PX_TQSTAT in a future release.

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
BYTES	NUMBER	The number of bytes produced/consumed
OPEN_TIME	NUMBER	Time (secs) the table queue remained open
AVG_LATENCY	NUMBER	Time (ms) 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	Operating system process identifier
USERNAME	VARCHAR2	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	Operating system terminal identifier
PROGRAM	VARCHAR2	Program in progress
BACKGROUND	VARCHAR2	1 for a background process; NULL for a normal process
LATCHWAIT	VARCHAR2	Address of latch the process is waiting for; NULL if none
LATCHSPIN	VARCHAR2	Address of latch the process is being spun on; NULL if none

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; not needed for restore
DELETED	VARCHAR2(3)	If set to YES, indicates the copy is deleted, otherwise set to NO
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
FIRST_CHANGE#	NUMBER	First change number in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change number in the next log
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log in blocks
BLOCK_SIZE	NUMBER	Redo log block size
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds

V$PROXY_DATAFILE

This view contains descriptions of datafile and controlfile backups which are taken with a new feature called 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
DELETED	VARCHAR2(3)	If set to YES, indicates the copy is deleted, otherwise set to NO
FILE#	NUMBER	Absolute datafile number, or 0 if this is a controlfile 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
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

Column	Datatype	Description
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 controlfile backup), the RECID of the oldest offline range record in this controlfile 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

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
SYSDBA	VARCHAR2(5)	If the value of this column is TRUE, the user can connect with SYSDBA privileges
SYSOPER	VARCHAR2(5)	If the value of this column is 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 parallel server (P000, P001, etc)
STATUS	VARCHAR2(9)	The state of the parallel server. Either In Use or 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
VALUE	NUMBER	The value of the statistic

The STATISTIC column's values are:

Table 3–9 Names of Statistics in the STATISTIC Column

Statistic (Fixed Row)	Description
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 increase performance due to avoiding the latency of PX server process creation

Table 3–9 Names of Statistics in the STATISTIC Column

Statistic (Fixed Row)	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, it is recommended that the cause be determined
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

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
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this parallel server process belongs

Column	Datatype	Description
SERVER_SET	NUMBER	The logical set of servers that this parallel server process belongs to. A single server group will have at most two server sets
SERVER#	NUMBER	The logical number of a parallel server 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 parallel server process belongs
SERVER_SET	NUMBER	The logical set of servers that this parallel server process belongs to. A single server group will have at most two server sets
SERVER#	NUMBER	The logical number of a parallel server 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$RECOVER_FILE

This view displays the status of files needing media recovery.

Column	Datatype	Description
FILE#	NUMBER	File identifier number
ONLINE	VARCHAR2	Online status: ONLINE, OFFLINE
ERROR	VARCHAR2	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$RECOVERY_FILE_STATUS contains one row for each datafile for each RECOVER command. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_FILE_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
FILENUM	NUMBER	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. Contains one of the following values: IN RECOVERY; CURRENT; NOT RECOVERED

For further information, see *Oracle8i Backup and Recovery Guide*.

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$LOG_HISTORY. For more information, see "V$LOG_HISTORY" on page 3-66.

V$RECOVERY_LOG contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_LOG will be empty to all other Oracle users.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
TIME	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"

For further information, see *Oracle8i Backup and Recovery Guide*.

V$RECOVERY_PROGRESS

V$RECOVERY_PROGRESS can be used to track database recovery operations to ensure that they are not stalled, and also to estimate the time required to complete the operation in progress.

V$RECOVERY_PROGRESS is a subview of V$SESSION_LONGOPS.

Column	Datatype	Description
TYPE	VARCHAR2(64)	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

For further information, see *Oracle8i Backup and Recovery Guide*.

V$RECOVERY_STATUS

V$RECOVERY_STATUS contains statistics of the current recovery process. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
RECOVERY_CHECKPOINT	DATE	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. Contains one of the following values: NEED LOG; LOG REUSED; THREAD DISABLED

For further information, see *Oracle8i Backup and Recovery Guide*.

V$REQDIST

This view lists statistics for the histogram of MTS 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

Column	Datatype	Description
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 3–3.
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 3–10.

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 3–10)
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 3–10 Values for RESOURCE_NAME column

Resource Name	Corresponds to this Initialization Parameter
DISTRIBUTED_TRANSACTIONS	DISTRIBUTED_TRANSACTIONS For more information on this parameter, see "DISTRIBUTED_TRANSACTIONS" on page 1-37.
DML_LOCKS	DML_LOCKS For more information on this parameter see "DML_LOCKS" on page 1-37.
ENQUEUE_LOCKS	This value is computed by Oracle. Use the V$ENQUEUE_LOCK view (described on on page 3-41) to obtain more information about the enqueue locks.
ENQUEUE_RESOURCES	ENQUEUE_RESOURCES For more information on this parameter see "ENQUEUE_RESOURCES" on page 1-39.
LM_PROCESSES	LM_PROCS For more information on this parameter see "LM_PROCS" on page 1-57.
LM_RESOURCES	LM_RESS For more information on this parameter see "LM_RESS" on page 1-57.
LM_LOCKS	LM_LOCKS For more information on this parameter see "LM_LOCKS" on page 1-56.
MTS_MAX_SERVERS	MTS_MAX_SERVERS For more information on this parameter see "MTS_MAX_SERVERS" on page 1-75.
PARALLEL_SLAVES	PARALLEL_MAX_SERVERS For more information on this parameter, see "PARALLEL_MAX_SERVERS" on page 1-96.

Table 3–10 Values for RESOURCE_NAME column

Resource Name	Corresponds to this Initialization Parameter
PROCESSES	PROCESSES For more information on this parameter, see "PROCESSES" on page 1-103.
ROLLBACK_SEGMENTS	MAX_ROLLBACK_SEGMENTS For more information on this parameter, see "MAX_ROLLBACK_SEGMENTS" on page 1-72.
SESSIONS	SESSIONS For more information on this parameter, see "SESSIONS" on page 1-114
SORT_SEGMENT_LOCKS	This value is computed by Oracle
TEMPORARY_LOCKS	This value is computed by Oracle
TRANSACTIONS	TRANSACTIONS For more information on this parameter, see "TRANSACTIONS" on page 1-125

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
EXTENTS	NUMBER	Number of extents in rollback segment
RSSIZE	NUMBER	Size in bytes of rollback segment
WRITES	NUMBER	Number of bytes written to 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 rollback segment

Column	Datatype	Description
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
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
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

Column	Datatype	Description
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
INST_LOCK_REQUEST	NUMBER	Relevant only for Parallel Server. The mode instance lock is being requested in
INST_LOCK_RELEASE	NUMBER	Relevant only for Parallel Server. Whether the instance lock needs to be released
INST_LOCK_TYPE	VARCHAR2(2)	Relevant only for Parallel Server. The type of instance lock
INST_LOCK_ID1	RAW(4)	Relevant only for Parallel Server. The ID associated with the instance lock
INST_LOCK_ID2	RAW(4)	Relevant only for Parallel Server. The ID associated with the instance lock
KEY	RAW(100)	Relevant only for Parallel Server. 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 the currently active resource consumer groups.

Column	Datatype	Description
NAME	VARCHAR2(32)	The name of the consumer group
ACTIVE_SESSIONS	NUMBER	The number of currently active sessions in this consumer group
EXECUTION_WAITERS	NUMBER	The number of currently active sessions waiting for an execution quantum
REQUESTS	NUMBER	The total number of requests that were executed in this consumer group
CPU_WAIT_TIME	NUMBER	The total amount of time that sessions waited for CPU
CPU_WAITS	NUMBER	The number of times all sessions in this consumer group had to wait for CPU
CONSUMED_CPU_TIME	NUMBER	The total amount of CPU time consumed by all sessions in this consumer group
YIELDS	NUMBER	The number of times sessions in this consumer group had to yield the CPU
SESSIONS_QUEUED	NUMBER	The count of currently queued sessions waiting to become active

V$RSRC_CONSUMER_GROUP_CPU_MTH

This view shows all available resource allocation methods for resource consumer groups.

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.

Column	Datatype	Description
NAME	VARCHAR2(32)	The name of the resource plan

V$RSRC_PLAN_CPU_MTH

This view shows all available CPU resource allocation methods for resource plans.

Column	Datatype	Description
NAME	VARCHAR2(32)	The name of the resource allocation method

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

Column	Datatype	Description
COMMAND	NUMBER	Command in progress (last statement parsed); for a list of values, see Table 3–7
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 4Byte value. The low-order 2Bytes 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)
SERVER	VARCHAR2(9)	Server type: DEDICATED, SHARED, PSEUDO, NONE
SCHEMA#	NUMBER	Schema user identifier
SCHEMANAME	VARCHAR2(30)	Schema user name
OSUSER	VARCHAR2(15)	Operating system client user name
PROCESS	VARCHAR2(9)	Operating system client process ID
MACHINE	VARCHAR2(64)	Operating system machine name
TERMINAL	VARCHAR2(10)	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
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

Column	Datatype	Description
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
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_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.
PDML_ENABLED	VARCHAR2(3)	This column has been replaced by PDML_ENABLED and PDML_ STATUS. See above.
FAILOVER_TYPE	VARCHAR2(10)	NONE if failover is disabled for this session, SESSION if client is able to failover its session following a disconnect, and SELECT if client is able to fail over selects in progress as well
FAILOVER _METHOD	VARCHAR2(3)	NONE if failover is disabled for this session, BASIC if client reconnects following a disconnect, PRECONNECT if the backup instance is able to support all connections from every instance that it is backup for

Column	Datatype	Description
FAILED_OVER	VARCHAR2(13)	TRUE if running in failover mode and have failed over, otherwise FALSE
RESOURCE_CONSUMER_GROUP	VARCHAR2(32)	Name of the session's current resource consumer group

Table 3–11 lists numeric values corresponding to commands that may be in progress during a session. These values can appear in the V$SESSION COMMAND column. They also appear in the data dictionary view SYS.AUDIT_ACTIONS.

Table 3–11 Command Numbers for the COMMAND Column

Command Number	Command
0	No command in progress. Occurs when process is in a transitory state, usually when terminating.
1	CREATE TABLE
2	INSERT
3	SELECT
4	CREATE CLUSTER
5	ALTER CLUSTER
6	UPDATE
7	DELETE
8	DROP CLUSTER
9	CREATE INDEX
10	DROP INDEX
11	ALTER INDEX
12	DROP TABLE
13	CREATE SEQUENCE
14	ALTER SEQUENCE
15	ALTER TABLE
16	DROP SEQUENCE

Table 3–11 Command Numbers for the COMMAND Column

Command Number	Command
17	GRANT
18	REVOKE
19	CREATE SYNONYM
20	DROP SYNONYM
21	CREATE VIEW
22	DROP VIEW
23	VALIDATE INDEX
24	CREATE PROCEDURE
25	ALTER PROCEDURE
26	LOCK TABLE
27	NO OPERATION
28	RENAME
29	COMMENT
30	AUDIT
31	NOAUDIT
32	CREATE DATABASE LINK
33	DROP DATABASE LINK
34	CREATE DATABASE
35	ALTER DATABASE
36	CREATE ROLLBACK SEGMENT
37	ALTER ROLLBACK SEGMENT
38	DROP ROLLBACK SEGMENT
39	CREATE TABLESPACE
40	ALTER TABLESPACE
41	DROP TABLESPACE
42	ALTER SESSION
43	ALTER USER

Table 3–11 Command Numbers for the COMMAND Column

Command Number	Command
44	COMMIT
45	ROLLBACK
46	SAVEPOINT
47	PL/SQL EXECUTE
48	SET TRANSACTION
49	ALTER SYSTEM SWITCH LOG
50	EXPLAIN
51	CREATE USER
52	CREATE ROLE
53	DROP USER
54	DROP ROLE
55	SET ROLE
56	CREATE SCHEMA
57	CREATE CONTROL FILE
58	ALTER TRACING
59	CREATE TRIGGER
60	ALTER TRIGGER
61	DROP TRIGGER
62	ANALYZE TABLE
63	ANALYZE INDEX
64	ANALYZE CLUSTER
65	CREATE PROFILE
66	DROP PROFILE
67	ALTER PROFILE
68	DROP PROCEDURE
69	DROP PROCEDURE

Table 3–11 Command Numbers for the COMMAND Column

Command Number	Command
70	ALTER RESOURCE COST
71	CREATE SNAPSHOT LOG
72	ALTER SNAPSHOT LOG
73	DROP SNAPSHOT LOG
74	CREATE SNAPSHOT
75	ALTER SNAPSHOT
76	DROP SNAPSHOT
79	ALTER ROLE
85	TRUNCATE TABLE
86	TRUNCATE CLUSTER
88	ALTER VIEW
91	CREATE FUNCTION
92	ALTER FUNCTION
93	DROP FUNCTION
94	CREATE PACKAGE
95	ALTER PACKAGE
96	DROP PACKAGE
97	CREATE PACKAGE BODY
98	ALTER PACKAGE BODY
99	DROP PACKAGE BODY

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)

Column	Datatype	Description
AUTHENTICATION_ TYPE	VARCHAR2(15)	How the user was authenticated: OS, PROTOCOL, or NETWORK.
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. For more information, see "TIMED_STATISTICS" on page 1-124.

Column	Datatype	Description
SID	NUMBER	The ID of the session
EVENT	VARCHAR2(64)	The name of the wait event. For more information, see 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 certain long-running operations. It provides progression reports on operations using the columns SOFAR and TOTALWORK. For example, the operational status for the following components can be monitored:

- hash cluster creations
- backup operations
- recovery operations

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
OPNAME	VARCHAR2(64)	The operation name
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

Column	Datatype	Description
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds from the start of operations
CONTEXT	NUMBER	Context
MESSAGE	VARCHAR2(512)	Statistics summary message

V$SESSION_OBJECT_CACHE

This view displays object cache statistics for the current user session on the local server (instance).

Column	Datatype	Description
PINS	NUMBER	Number of object pins or look-ups in the cache
HITS	NUMBER	Number of object pins that found the object already in the cache
TRUE_HITS	NUMBER	Number of object pins that found the object already in the cache and in the desired state (thus, not requiring refresh from the database)
HIT_RATIO	NUMBER	The ratio of 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. For more information, see "TIMED_ STATISTICS" on page 1-124.

  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.

For more information on session waits, 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. For more information, see 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

Column	Datatype	Description
WAIT_TIME	NUMBER	A non-zero 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 (see Table 3–12)

Table 3–12 defines values in the V$SESSION_WAIT STATE column.

Table 3–12 Wait State listed in the STATE Column

STATE	Value	Meaning
WAITING	0	The session is currently waiting
WAITED UNKNOWN TIME	-2	Duration of last wait is unknown
WAITED SHORT TIME	-1	Last wait < 1/100th of a second
WAITED KNOWN TIME	>0	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#), see "V$STATNAME" on page 3-115.

Column	Datatype	Description
SID	NUMBER	Session identifier
STATISTIC#	NUMBER	Statistic number (identifier)
VALUE	NUMBER	Statistic value

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$SGA

This view contains summary information on the System Global Area.

Column	Datatype	Description
NAME	VARCHAR2	SGA component group
VALUE	NUMBER	Memory size in bytes

V$SGASTAT

This view contains detailed information on the System Global Area.

Column	Datatype	Description
NAME	VARCHAR2	SGA component name
BYTES	NUMBER	Memory size in bytes
POOL	VARCHAR2	Designates the pool in which the memory in NAME resides. Value can be LARGE POOL - Memory is allocated from the large pool or SHARED POOL - Memory is allocated from the shared pool

V$SHARED_POOL_RESERVED

This fixed view lists statistics that help you tune the reserved pool and space within the shared pool.

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. For more information, see "SHARED_POOL_RESERVED_SIZE" on page 1-115.

Column	Datatype	Description
FREE_SPACE	NUMBER	Total amount of free space on the reserved list
AVG_FREE_SIZE	NUMBER	Average size of the free memory on the reserved list
FREE_COUNT	NUMBER	Number of free pieces of memory on the reserved list
MAX_FREE_SIZE	NUMBER	Size of the largest free piece of memory on the reserved list
USED_SPACE	NUMBER	Total amount of used memory on the reserved list
AVG_USED_SIZE	NUMBER	Average size of the used memory on the reserved list
USED_COUNT	NUMBER	Number of used pieces of memory on the reserved list
MAX_USED_SIZE	NUMBER	Size of the largest used piece of memory on the reserved list
REQUESTS	NUMBER	Number of times that the reserved list was searched for a free piece of memory
REQUEST_MISSES	NUMBER	Number of times the reserved list did not have a free piece of memory to satisfy the request, and started flushing objects from the LRU list
LAST_MISS_SIZE	NUMBER	Request size of the last request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list
MAX_MISS_SIZE	NUMBER	Request size of the largest request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list

The following columns of V$SHARED_POOL_RESERVED contain values which are valid even if SHARED_POOL_RESERVED_SIZE is not set.

Column	Datatype	Description
REQUEST_FAILURES	NUMBER	Number of times that no memory was found to satisfy a request (that is, the number of times the error ORA-4031 occurred)
LAST_FAILURE_SIZE	NUMBER	Request size of the last failed request (that is, the request size for the last ORA-4031 error)

Column	Datatype	Description
ABORTED_REQUEST _ THRESHOLD	NUMBER	Minimum size of a request which signals an ORA-4031 error without flushing objects
ABORTED_REQUESTS	NUMBER	Number of requests that signalled an ORA-4031 error without flushing objects
LAST_ABORTED_SIZE	NUMBER	Last size of the request that returned an ORA-4031 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
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$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

Column	Datatype	Description
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
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$SORT_USAGE

This view describes sort usage.

Column	Datatype	Description
USER	VARCHAR2(30)	User who requested temporary space
SESSION_ADDR	RAW(4)	Address of shared SQL cursor
SESSION_NUM	NUMBER	Serial number of session
SQLADDR	RAW(4)	Address of SQL statement

Column	Datatype	Description
SQLHASH	NUMBER	Hash value of SQL statement
TABLESPACE	VARCHAR2(31)	Tablespace in which space is allocated
CONTENTS	VARCHAR2(9)	Indicates whether tablespace is TEMPORARY / PERMANENT
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$SQL

This view 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)	The first eighty characters of the SQL text for the current cursor
SHARABLE_MEM	NUMBER	The amount of sharable memory, in bytes used by this child cursor
PERSISTENT_MEM	NUMBER	The amount of persistent memory, in bytes used by this child cursor
RUNTIME_MEM	NUMBER	The size of the ephemeral frame used by this child cursor
SORTS	NUMBER	The number of sorts that was done for this child cursor
LOADED_VERSIONS	NUMBER	1 if context heap is loaded, 0 otherwise
OPEN_VERSIONS	NUMBER	1 if the child cursor is locked, 0 otherwise
USERS_OPENING	NUMBER	The number of users executing the statement
EXECUTIONS	NUMBER	The number of executions that took place on this object since it was brought into the library cache
USERS_EXECUTING	NUMBER	The number of users executing the statement
LOADS	NUMBER	The number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	The time stamp of the parent creation time
INVALIDATIONS	NUMBER	The number of times this child cursor has been invalidated
PARSE_CALLS	NUMBER	The number of parse calls for this child cursor

Column	Datatype	Description
DISK_READS	NUMBER	The number of disk reads for this child cursor
BUFFER_GETS	NUMBER	The number of buffer gets for this child cursor
ROWS_PROCESSED	NUMBER	The total number of rows the parsed SQL statement returns
COMMAND_TYPE	NUMBER	The Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement is executed
OPTIMIZER_COST	NUMBER	The cost of this query given by the optimizer
PARSING_USER_ID	NUMBER	The user ID of the user who originally built this child cursor
PARSING_SCHEMA_ID	NUMBER	The 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 cache using the DBMS_SHARED_POOL package
ADDRESS	RAW(4)	The address of the handle to the parent for this cursor
TYPE_CHK_HEAP	RAW(4)	The descriptor of the type check heap for this child cursor
HASH_VALUE	NUMBER	The hash value of the parent statement in the library cache
CHILD_NUMBER	NUMBER	The 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 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
SERIALIZABLE_ ABORTS	NUMBER	The number of times the transaction fails to serialize, producing ORA-8177 errors, per cursor

V$SQL_BIND_DATA

This view displays the actual bind data sent by the client for each distinct bind variable in each cursor owned by the session querying this view if the data is available in the server.

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

This view displays bind metadata provided by the client for each distinct bind variable in each cursor owned by the session querying this view.

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

Column	Datatype	Description
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)	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
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_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 that this row is displaying information for
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)	This is the whole comment field that was supplied when this memory chunk was allocated
CHUNK_PTR	RAW(4)	This is 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 LRU'd out of the shared pool
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$SQLAREA

This view 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)	The first eighty characters of the SQL text for the current cursor
SHARABLE_MEM	NUMBER	The sum of all sharable memory, in bytes, of all the child cursors under this parent
PERSISTENT_MEM	NUMBER	The sum of all persistent memory, in bytes, of all the child cursors under this parent

Column	Datatype	Description
RUNTIME_MEM	NUMBER	The sum of all the ephemeral frame sizes of all the children
SORTS	NUMBER	The sum of the number of sorts that was done for all the children
VERSION_COUNT	NUMBER	The number of children that are present in the cache under this parent
LOADED_VERSIONS	NUMBER	The number of children 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
EXECUTIONS	NUMBER	The total number of executions, totalled over all the children
USERS_EXECUTING	NUMBER	The total number of users executing the statement over all children
LOADS	NUMBER	The number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	The time stamp of the parent creation time
INVALIDATIONS	NUMBER	The total number of invalidations over all the children
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

Column	Datatype	Description
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
SERIALIZABLE_ABORTS	NUMBER	The number of times the transaction fails to serialize, producing ORA-8177 errors, totalled over all the children

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, etc.)

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. For more information, see "V$SQLTEXT" on page 3-114.

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, etc.)

V$STATNAME

This view displays decoded statistic names for the statistics shown in the V$SESSTAT and V$SYSSTAT tables. For more information, see "V$SESSTAT" on page 3-103 and "V$SYSSTAT" on page 3-119.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2	Statistic name. See also Table 3–13
CLASS	NUMBER	Statistic class: 1 (User); 2 (Redo); 4 (Enqueue); 8 (Cache); 16 (OS), 32 (Parallel Server); 64 (SQL); 128 (Debug)

Table 3–13 lists the generic Oracle Server statistics returned by V$STATNAME. For a complete description of each statistic, see Appendix C, "Statistics Descriptions".

Table 3–13 V$SESSTAT and V$SYSSTAT Statistics Names

CPU used by this session	CPU used when call started
CR blocks created	Cached Commit SCN referenced
Commit SCN referenced	Current blocks converted for CR
DBWR buffers scanned	DBWR checkpoint buffers written
DBWR checkpoints	DBWR forced writes
DBWR free buffers found	DBWR lru scans
DBWR make free requests	DBWR revisited being-written buffer
DBWR skip hot writes	DBWR summed scan depth
DBWR transaction table writes	DBWR undo block writes
DDL statements parallelized	DFO trees parallelized
DML statements parallelized	OS All other sleep time
OS Chars read and written	OS Data page fault sleep time
OS Input blocks	OS Involuntary context switches
OS Kernel page fault sleep time	OS Major page faults
OS Messages received	OS Messages sent
OS Minor page faults	OS Other system trap CPU time
OS Output blocks	OS Process heap size

Table 3–13 V$SESSTAT and V$SYSSTAT Statistics Names

OS Process stack size	OS Signals received
OS Swaps	OS System call CPU time
OS System calls	OS Text page fault sleep time
OS User level CPU time	OS User lock wait sleep time
OS Voluntary context switches	OS Wait-cpu (latency) time
PX local messages received	PX local messages sent
PX remote messages received	PX remote messages sent
Parallel operations downgraded to serial	SQL*Net roundtrips to/from client
SQL*Net roundtrips to/from dblink	Unnecessary process cleanup for SCN batching
background checkpoints completed	background checkpoints started
background timeouts	buffer is not pinned count
buffer is pinned count	bytes received via SQL*Net from client
bytes received via SQL*Net from dblink	bytes sent via SQL*Net to client
bytes sent via SQL*Net to dblink	calls to get snapshot scn: kcmgss
calls to kcmgas	calls to kcmgcs
calls to kcmgrs	change write time
cleanouts and rollbacks - consistent read gets	cleanouts only - consistent read gets
cluster key scan block gets	cluster key scans
commit cleanout failures: block lost	commit cleanout failures: buffer being written
commit cleanout failures: callback failure	commit cleanout failures: cannot pin
commit cleanout failures: hot backup in progress	commit cleanout failures: write disabled
commit cleanouts	commit cleanouts successfully completed
consistent changes	consistent gets
cursor authentications	db block changes
db block gets	deferred (CURRENT) block cleanout applications
dirty buffers inspected	enqueue conversions
enqueue deadlocks	enqueue releases
enqueue requests	enqueue timeouts
enqueue waits	exchange deadlocks

Table 3–13 V$SESSTAT and V$SYSSTAT Statistics Names

execute count	free buffer inspected
free buffer requested	global cache convert time
global cache convert timeouts	global cache converts
global cache cr block receive time	global cache cr blocks received
global cache cr read from disk	global cache cr timeouts
global cache defers	global cache fairness down converts
global cache freelist waits	global cache get time
global cache convert time	global cache queued converts
global lock async converts	global lock async gets
global lock convert time	global lock converts (async)
global lock converts (non async)	global lock get time
global cache hash latch waits	global lock gets (async)
global lock gets (non async)	global lock releases
global lock sync converts	global lock sync gets
hot buffers moved to head of LRU	immediate (CR) block cleanout applications
immediate (CURRENT) block cleanout applications	instance recovery database freeze count
kcmccs called get current scn	kcmccs read scn without going to DLM
kcmccs waited for batching	logons cumulative
logons current	messages received
messages sent	native hash arithmetic execute
native hash arithmetic fail	next scns gotten without going to DLM
no buffer to keep pinned count	no work-consistent read gets
opened cursors cumulative	opened cursors current
opens of replaced files	opens requiring cache replacement
parse count (hard)	parse count (total)
parse time cpu	parse time elapsed
physical reads	physical writes
physical reads direct	physical writes direct
physical writes non-checkpoint	physical writes non-checkpoint

Table 3–13 V$SESSTAT and V$SYSSTAT Statistics Names

pinned buffers inspected	process last non-idle time
queries parallelized	recovery array read time
recovery array reads	recovery blocks read
recursive calls	recursive cpu usage
redo blocks written	redo buffer allocation retries
redo entries	redo log space requests
redo log space wait time	redo log switch interrupts
redo ordering marks	redo size
redo synch time	redo sync writes
redo wastage	redo write time
redo writer latching time	redo writes
remote instance undo block writes	remote instance undo header writes
rollback changes-undo records applied	rollbacks only-consistent read gets
serializable aborts	session connect time
session cursor cache count	session cursor cache hits
session logical reads	session pga memory
session pga memory max	session stored procedure space
session uga memory	session uga memory max
sorts (disk)	sorts (memory)
sorts (rows)	summed dirty queue length
table fetch by rowid	table fetch continued row
table scan blocks gotten	table scan rows gotten
table scans (cache partitions)	table scans (direct read)
table scans (long tables)	table scans (rowid ranges)
table scans (short tables)	total file opens
transaction lock background get time	transaction lock background gets
transaction lock foreground requests	transaction lock foreground wait time

Table 3–13 V$SESSTAT and V$SYSSTAT Statistics Names

transaction rollbacks	transaction tables consistent read rollbacks
transaction tables consistent reads-undo records applied	user calls
user commits	user rollbacks

Additional Information: On some platforms, the NAME and CLASS columns will contain additional operating system-specific statistics. See your operating system-specific Oracle documentation for more information about these statistics.

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#), see "V$STATNAME" on page 3-115.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Statistic name. See Table 3-13 on page 107.

Column	Datatype	Description
CLASS	NUMBER	Statistic class: 1 (User); 2 (Redo); 4 (Enqueue); 8 (Cache); 16 (OS), 32 (Parallel Server); 64 (SQL); 128 (Debug)
VALUE	NUMBER	Statistic value

V$SYSTEM_CURSOR_CACHE

This view displays similar information to the V$SESSION_CURSOR_CACHE view except that this information is system wide. For more information, see "V$SESSION_CURSOR_CACHE" on page 3-99.

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. Please remember that doing this will have a small negative effect on system performance. For more information, see"TIMED_STATISTICS" on page 1-124.

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

This view contains information on system parameters.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(64)	Parameter name
TYPE	NUMBER	Parameter type; 1 = Boolean, 2 = string, 3 = integer
VALUE	VARCHAR2(512)	Value assigned to the parameter
ISDEFAULT	VARCHAR2(9)	Is the value assigned to the parameter the default
ISSES_MODIFIABLE	VARCHAR2(5)	Whether the parameter can be modified by ALTER SESSION
ISSYS_MODIFIABLE	VARCHAR2(9)	Whether the parameter can be modified by ALTER SYSTEM
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SESSION was performed, the value will be MODIFIED. If an ALTER SYSTEM (which will cause all the currently logged in sessions' values to be modified) was performed the value will be SYS_ MODIFIED
ISADJUSTED	VARCHAR2(5)	Indicates that the rdbms adjusted the input value to a more suitable value (e.g., the parameter value should be prime, but the user input a non-prime number, so the rdbms adjusted the value to the next prime number)
DESCRIPTION	VARCHAR2(64)	Descriptive text about the parameter

V$TABLESPACE

This view displays tablespace information from the controlfile.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
NAME	VARCHAR2 (30)	Tablespace name

V$TEMPFILE

This view displays tempfile information.

Column	Datatype	Description
FILE#	NUMBER	The absolute file number

Column	Datatype	Description
CREATION_CHANGE#	NUMBER	The creation System Change Number
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
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$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)

Column	Datatype	Description
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
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_PING

The view V$TEMP_PING 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
FREQUENCY	NUMBER	The frequency
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	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_WRITE	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 _WRITE	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 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
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

Column	Datatype	Description
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file
OP_2_SS	NUMBER	Number of pcm locks ss locks opened. 0 in Oracle 8.1

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 per 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$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

Column	Datatype	Description
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
AVGIOTIM	NUMBER	The average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
LSTIOTIM	NUMBER	The time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MINIOTIM	NUMBER	The 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	The maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIORTM	NUMBER	The 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

Column	Datatype	Description
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$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

Column	Datatype	Description
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
NOUNDO	VARCHAR2(3)	"Yes" if a no undo transaction
PTX	VARCHAR 2(3)	YES if parallel transaction, otherwise set to NO
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 = transaction enqueue.
ID1	NUMBER	Lock identifier #1 (depends on type)

Column	Datatype	Description
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)
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$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$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. The following topic is included in this chapter:

- Database Limits

Database Limits

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. For more information on the maximum value of such limits, see your operating system-specific Oracle documentation.

Table 4–1 lists datatype limits

Table 4–2 lists physical database limits

Table 4–3 lists logical database limits

Table 4–4 lists process / runtime limits

Table 4–1 Datatype Limits

Datatypes	Limit	Comments
BFILE	maximum size: 4 GB maximum size of file name: 255 characters maximum size of directory name: 30 characters maximum number of open BFILEs: see comments	The maximum number of BFILEs is limited by the value of SESSION_MAX_OPEN_FILES, which is itself limited by the maximum number of open files the operating system will allow.
BLOB	4 GB maximum	The number of LOB columns per table is limited only by the maximum number of columns per table (i.e., 1000)
CHAR	2000 bytes maximum	
CHAR VARYING	4000 bytes	
CLOB	4 GB maximum	The number of LOB columns per table is limited only by the maximum number of columns per table (i.e., 1000)
Literals (characters or numbers in SQL or PL / SQL)	4000 characters maximum	
LONG	2^{31}-1 bytes (2 GB) maximum	Only one LONG column allowed per table
NCHAR	2000 bytes	
NCHAR VARYING	4000 bytes	

Table 4–1 Datatype Limits

Datatypes	Limit	Comments
NCLOB	4 GB maximum	The number of LOB columns per table is limited only by the maximum number of columns per table (i.e., 1000)
NUMBER	999...(38 9's) x10^{125}maximum value	Can be represented to full 38-digit precision (the mantissa).
	-999...(38 9's) x10^{125}minimum value	Can be represented to full 38-digit precision (the mantissa).
Precision	38 significant digits	
RAW	2000 bytes maximum	
VARCHAR	4000 bytes maximum	
VARCHAR2	4000 bytes maximum	

Table 4–2 Physical Database Limits

Item	Type of Limit	Limit Value
Database Block Size	minimum	2048 bytes; must be a multiple of O/S physical block size
	maximum	O/S-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 controlfiles	1 minimum: 2 or more (on separate devices) strongly recommended
	size of a controlfile	dependent on O/S and database creation options; maximum of 20,000 x (database block size)
Database files	maximum per tablespace	O/S 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 init parameter for a particular instance
Database file size	maximum	O/S dependent, limited by maximum O/S file size; typically 2^{22} or 4M blocks

Table 4–2 Physical Database Limits

Item	Type of Limit	Limit Value
MAXEXTENTS	default value	derived from tablespace default storage or DB_BLOCK_SIZE
	maximum	unlimited
Redo Log Files	maximum number of logfiles	LOG_FILES initialization parameter, or MAXLOGFILES in CREATE DATABASE; controlfile can be resized to allow more entries; ultimately an O/S limit
	maximum number of logfiles per group	unlimited
Redo Log File Size	minimum size	50K bytes
	maximum size	O/S limit, typically 2GB
Tablespaces	maximum number per database	64K Number of tablespaces cannot exceed the number of database files, as each tablespace must include at least one file.

Table 4–3 Logical Database Limits

Item	Type	Limit
GROUP BY clause	maximum length	The group-by expression and all of the non-distinct aggregates (e.g., sum, avg) need to fit within a single database block.
Indexes	maximum per table	unlimited
	total size of indexed column	40% of the database block size minus some overhead.
Columns	table	1000 columns maximum
	indexed (or clustered index)	32 columns maximum
	bitmapped index	30 columns maximum
Constraints	maximum per column	unlimited
Nested Queries	maximum number	255

Table 4–3 Logical Database Limits

Item	Type	Limit
Partitions	maximum length of linear partitioning key	4KB - overhead
	maximum number of columns in partition key	16 columns
	maximum number of partitions allowed per table or index	64K-1 partitions
Rollback Segments	maximum number per database	no limit; limited within a session by MAX_ ROLLBACK_SEGMENTS init parameter
Rows	maximum number per table	no limit
SQL Statement Length	maximum length of statements	64K 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. Consult your PL/SQL or Developer/2000 documentation for details. The limits typically range from 2000-3000 lines of code.
Trigger Cascade Limit	maximum value	O/S dependent, typically 32
Users and Roles	maximum	2,147,483,638
Tables	maximum per clustered table	32 tables
	maximum per database	unlimited

Table 4–4 Process / Runtime Limits

Item	Type	Limit
Instances per database	maximum number of OPS instances per database	O/S dependent
Locks	row-level	unlimited
	Distributed Lock Manager	O/S dependent
SGA size	maximum value	O/S dependent, typically 2-4 GB for 32-bit O/S, > 4 GB for 64 bit O/S
Advanced Queuing Processes	maximum per instance	10
Job Queue Processes	maximum per instance	36

Table 4–4 Process / Runtime Limits

Item	Type	Limit
I/O Slave Processes	maximum per background process (DBWR, LGWR, etc.)	15
	maximum per Backup session	15
Sessions	maximum per instance	32K, limited by PROCESSES and SESSIONS init parameters
LCK Processes	maximum per instance	10
MTS Servers	maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS init parameters, for instance.
Dispatchers	maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS init parameters, for instance.
Parallel Execution Slaves	maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS init parameters, for instance.
Backup Sessions	maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS init parameters, for instance.

5

SQL Scripts

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

The SQL scripts are described in the following sections:

- Creating the Data Dictionary
- Creating Additional Data Dictionary Structures
- The "NO" Scripts
- Migration Scripts

Note: Check the header of each SQL script for more detailed information and examples.

Creating the Data Dictionary

The data dictionary is automatically created when a database is created. 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 and must remain in the SYSTEM tablespace. The data dictionary base tables are present to store information about all user-defined objects in the database.

Table 5–1 lists the scripts that are required for the Oracle Server with the indicated options. The appropriate scripts for your Oracle Server options are run automatically when you create a database. They are described here because you might need to run them again, when upgrading to a new release of Oracle8*i*. Your release notes and *Oracle8i Migration* indicate when this is necessary. Run these scripts connected to the Oracle Server as the user SYS.

The exact names and locations of these scripts are operating system dependent. See your operating system specific Oracle documentation for the names and locations on your system.

For more information about scripts with names starting with DBMS, see the *Oracle8i Administrator's Guide*.

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, and grants PUBLIC access to the synonyms
CATPROC.SQL	All databases	Runs all scripts required for or used with PL/SQL. It is required for all Oracle8*i* databases

Creating Additional Data Dictionary Structures

Oracle supplies other scripts with the Oracle Server that create additional structures you can use in managing your database and creating database applications. These scripts are listed in Table 5–2.

The exact names and locations of these scripts are operating system dependent. See your operating system-specific Oracle documentation for the names and locations on your system.

Table 5–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
CATBLOCK.SQL	Performance Management	Must be run when connected to SYS	Creates views that can dynamically display lock dependency graphs
CATEXP7.SQL	Exporting data to Oracle7	Must be run when connected to SYS	Creates the dictionary views needed for the Oracle7 Export utility to export data from Oracle8*i* in Oracle7 Export file format
CATHS.SQL	Heterogeneous Services	Must be run when connected to SYS	Installs packages for administering heterogeneous services
CATIO.SQL	Performance Management	Must be run when connected to SYS	Allows I/O to be traced on a table-by-table basis
CATOCTK.SQL	Security	Must be run when connected to SYS	Creates the Oracle Cryptographic Toolkit package
CATPARR.SQL	Parallel Server	SYS or SYSDBA	Creates parallel server data dictionary views
CATREP.SQL	Advanced Replication	Must be run when connected to 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

Table 5–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
DBMSOTRC.SQL	Performance Management	SYS or SYSDBA	Used to enable and disable Oracle Trace trace generation
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
UTLCHAIN1.SQL	Storage Management	any user	For use with Oracle 8.1. 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 CHECK constraints on date columns are year 2000 compliant
UTLDTREE.SQL	Metadata Management	any user	Creates tables and views that show dependencies between objects
UTLEXCPT1.SQL	Constraints	any user	For use with Oracle 8.1. Creates the default table (EXCEPTIONS) for storing exceptions from enabling constraints. Can handle both physical and logical rowids
UTLHTTP.SQL	Web Access	SYS or SYSDBA	PL/SQL package retrieve data from Internet or intranet web servers via HTTP protocol
UTLIP.SQL	PL/SQL	SYS	Used primarily for migration, upgrade, and downgrade operations. It invalidates all existing PL/SQL modules by altering certain dictionary tables so that subsequent recompilations will happen 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 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 certain dictionary tables. Then, it reloads the packages STANDARD and DBMS_STANDARD, which are necessary for using PL/SQL. Finally, it triggers a recompile of all PL/SQL modules, such as packages, procedures, types, etc.
UTLLOCKT.SQL	Performance Monitoring	SYS or SYSDBA	Displays a lock wait-for graph, in tree structure format
UTLPG.SQL	Data Conversion	SYS or SYSDBA	Provides a package that converts IBM/370 VS COBOL II
UTLPWDMG.SQL	Security	SYS or SYSDBA	Creates PL/SQL function 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, types, etc.
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	Advanced Replication	any user	Copies a snapshot schema from another snapshot site
UTLTKPROF.SQL	Performance Management	SYS	Creates the TKPROFER role to allow the TKPROF profiling utility to be runs by non-DBA users
UTLVALID.SQL	Partitioned Tables	any user	Creates table 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 command

The "NO" Scripts

The scripts in Table 5–3 are used to remove dictionary information for certain optional services or components.

Table 5–3 The NO Scripts

Script Name	Needed For	Run By	Description
CATNOADT.SQL	Objects	Must be run when connected to SYS	Drops views and synonyms on dictionary metadata that relate to Object types
CATNOAUD.SQL	Security	Must be run when connected to SYS	Drops views and synonyms on auditing metadata
CATNOHS.SQL	Heterogeneous Services	Must be run when connected to SYS	Removes Heterogeneous Services dictionary metadata
CATNOPRT.SQL	Partitioning	Must be run when connected to SYS	Drops views and synonyms on dictionary metadata that relate to partitioned tables and indexes
CATNOQUEUE .SQL	Advanced Queuing	Must be run when connected to SYS	Removes Advanced Queuing dictionary metadata
CATNORMN.SQL	Recovery Manager	Owner of recovery catalog	Removes recovery catalog schema
CATNOSVM.SQL	Server Manager	Must be run when connected to SYS	Removes Oracle7 Server Manager views and synonyms
CATNSNMP.SQL	Distributed Management	SYS	Drops the DBSNMP user and SNMPAGENT role

For more information, see *Oracle8i Migration*.

Migration Scripts

The scripts in Table 5–4 are useful when migrating to another version or release.

For more information, see *Oracle8i Migration*.

Table 5–4 Migration Scripts

Script Name	Needed For	Run By	Description
DROPCAT6.SQL	Removing legacy metadata	SYS	Drops the Oracle6 data dictionary catalog views
DROPCAT5.SQL	Removing legacy metadata	SYS	Drops the Oracle5 data dictionary catalog views
R070304.SQL	Replication	SYS	Performs a post-CATREP.SQL replication upgrade
RM804813.SQL	Recovery Manager	Owner of the Recovery Catalog Tables	Upgrades the recovery catalog from either 8.0.4 or 8.0.5 to 8.1.3
U703040.SQL	Migration from Oracle7	SYS or SYSDBA	Creates new Oracle8*i* dictionary metadata

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

The V$SESSION_WAIT view displays the events for which sessions have just completed waiting or are currently waiting. The V$SYSTEM_EVENT displays the total number of times all the sessions have waited for the events in that view.

The V$SESSION_EVENT is similar to V$SYSTEM_EVENT, but displays all waits for each session. For more information on these views, see "V$SESSION_EVENT" on page 3-99, "V$SESSION_WAIT" on page 3-102, and "V$SYSTEM_EVENT" on page 3-120.

Many of these wait events are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice. This may even happen during a bug fix patch release. 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 server. The columns P1, P2, and P3 represent parameters for the wait event.

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
BFILE check if exists			
BFILE check if open			
BFILE closure			
BFILE get length			
BFILE get name object			
BFILE get path object			
BFILE internal seek			
BFILE open			
BFILE read			

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
Contacting SCN server or SCN lock master			
DFS db file lock	file#	not used	not used
DFS lock handle	type | mode	id1	id2
DLM generic wait event			
IO clients wait for LMON to join GMS group			
LMON wait for LMD to inherit communication channels			
Null event			
PAR RECOV: Dequeue msg-Slave			
PAR RECOV: Wait for reply-Query Coord			
PL/SQL lock timer	duration	not used	not used
Parallel Execution Idle Wait-Slaves wait event			
Replication Dequeue			
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
Test if message present			
WMON goes to sleep	not used	not used	not used
Wait for a parallel reco to abort			
Wait for a undo record			

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
Wait for credit-free buffer			
Wait for credit-need buffer to send			
Wait for credit-send blocked			
Wait for slaves to ACK-Query Coord			
Wait for slaves to join-Query Coord			
Wait for stopper event to be increased			
alter system set mts_dispatcher	waited	not used	not used
batched allocate scn lock request	not used	not used	not used
buffer being modified waits			
buffer busy due to global cache	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
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
direct path write	file number	first dba	block cnt
dispatcher shutdown			

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
dispatcher timer	sleep time	not used	not used
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	not used	not used
kcl bg acks	count	loops	not used

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
kdi: Done Message Dequeue-Coord			
ktpr: Done Message Dequeue-Coord			
latch activity	address	number	process#
latch free	address	number	tries
library cache load lock	object address	lock address	10*mask+namespace
library cache lock	handle address	lock address	10*mode+namespace
library cache pin	handle address	pin address	10*mode+namespace
local write wait			
lock manager wait for dlmd to shutdown			
lock manager wait for remote message	waittime	not used	not used
log buffer space	not used	not used	not used
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)	not used	not used	not used
log file switch (checkpoint incomplete	not used	not used	not used
log file switch (clearing log file)	not used	not used	not used
log file switch completion	not used	not used	not used
log file sync	buffer#	not used	not used
log switch/archive	thread#	not used	not used
name-service call wait			
on-going SCN fetch to complete	not used	not used	not used
parallel execution create server	nservers	sleeptime	enqueue
parallel execution dequeue wait	reason	sleeptime/ senderid	passes
parallel execution qref latch	function	sleeptime	qref
parallel execution server shutdown	nalive	sleeptime	loop
parallel execution signal server	serial	error	nbusy
pending global transaction(s)	scans	not used	not used

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
pipe get	handle address	buffer length	timeout
pipe put	handle address	record length	timeout
pmon rdomain attach			
pmon timer	duration	not used	not used
process startup	type	process#	waited
queue messages	queue id	process#	wait time
rdbms ipc message	timeout	not used	not used
rdbms ipc message block	not used	not used	not used
rdbms ipc reply	from_process	timeout	not used
redo wait	not used	not used	not used
refresh controlfile command			
reliable message			
retry contact SCN lock master			
row cache lock	cache id	mode	request
scginq AST call	not used	not used	not used
secondary event			
single-task message	not used	not used	not used
slave exit	nalive	sleeptime	loop
slave wait			
smon timer	sleep time	failed	not used
sort segment request			
switch logfile command	not used	not used	not used
timer in sksawat	not used	not used	not used
trace continue			
trace unfreeze			
trace writer I/O			
trace writer flush			
transaction	undo seg#| slot#	wrap#	count
unbound tx	not used	not used	not used

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
undo segment extension	segment#	not used	not used
undo segment recovery	segment#	tx flags	not used
undo segment tx slot	segment#	not used	not used
virtual circuit status	circuit#	status	not used
wait for DLM latch	latchtype	gets	immediate
wait for DLM			
wait for DLM process allocation			
wait for DLM reconfiguration to complete			
wait for checking DLM domain			
wait for gms de-registration			
wait for gms registration			
wait for influx DLM latch	latchtype	latchaddr	not used
wait for lmd and pmon to attach DLM			
wait for lock db to become frozen			
wait for lock db to unfreeze			
wait for ownership of group-owned lock			
wait for pmon to exit			
wait for reconfiguration to start			
wait for recovery domain attach			
wait for recovery domain latch in kjpr			
wait for recovery validate to complete			
wait for register recovery to complete			
wait for send buffers to send DLM message			
wait for tickets to send DLM message			
wait to run in thread run			
waiting in scheduler enter			
waiting in scheduler enter2			

Table A–1 Wait Events for Oracle Parallel Server

Event Name	P1	P2	P3
waiting in scheduler shutdown			
waiting in scheduler system stop			
waiting in thread check			
waiting in thread check2			
waiting in thread end wait			
waiting in thread run (queued)			
write complete waits	file#	block#	id
writes stopped by instance recovery or database suspension	by thread #	our thread#	not used

Parameter Descriptions

This section describes a number of common event parameters.

- block#
- blocks
- break?
- class
- dba
- driver id
- file#
- id1
- id2
- lenum
- mode
- name and type
- namespace
- requests
- session#

- waited

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 these 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 non-zero 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, class 4 represents segment header.

dba

The initials "dba" represents the data block address. A dba 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–2 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 die.

Wait Event Descriptions

This section describes the Oracle events.

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-12.
waited	See "waited" on page A-12.

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-12.
waited	See "waited" on page A-12.

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-12.
waited	See "waited" on page A-12.

BFILE get length

The session waits on a call to check the size of an external large object (LOB).

Wait Time: The total elapsed time for the call to check the LOB size.

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

BFILE get name object

The session waits on a call to find or generate the external name of a external large object.

Wait Time: The total elapse time for **make external file name** to complete.

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

BFILE get path object

The session is waiting on a call to find or generate the external path name of an external large object (LOB).

Wait Time: The total elapsed time for **make external path** to complete.

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

BFILE internal seek

The session waits for a positioning call within the external large object (LOB) to complete.

Wait Time: The total elapse time for the **seek** to complete.

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

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-12.
waited	See "waited" on page A-12.

BFILE read

The session waits for a read from a external large object (LOB) to complete.

Wait Time: The total elapse time for the **read** to complete.

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

DFS db file lock

This event occurs only for the DBWR in the Parallel Server. 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, "Enqueue and Lock 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-11.

DFS lock handle

The session waits for the lock handle of a global lock request. The lock handle identifies a global lock. With this lock handle, other operations can be performed on this global lock (to identify the global lock in future operations such as conversions or release). The global lock is maintained by the DLM.

Wait Time: The session waits in a loop until it has obtained the lock handle from the DLM. Inside the loop there is a wait of 0.5 seconds.

Parameters:

name	See "name and type" on page A-12.
mode	See "mode" on page A-11.
id1	See "id1" on page A-11.
id2	See "id2" on page A-11.

The session needs to get the lock handle.

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.

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-10.
break?	See "break?" on page A-10.

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-10.
break?	See "break?" on page A-10.

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-10.
#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-10.
#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-10.
#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-10.
#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-10.

#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-10.
#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-10.
#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-10.
#bytes	The number of bytes that are sent over the database link to the other server process.

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

alter system set mts_dispatchers

A session has issued an alter system set mts_dispatchers=<string> and it 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	The number of times that the session has waited 1/100 of second.

batched allocate scn lock request

A session is waiting on another process to allocate an 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

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-11.
block#	See "block#" on page A-10.
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-10.
mode	See "mode" on page A-11.
flag	The flag points to the internal flags used by the session to get this block.
dba	See "dba" on page A-10.

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-10.
state	State refers to the status of the buffer contents.
mode	See "mode" on page A-11.

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 command 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 0xfffffff, the foreground waits on the LRU latch.

buffer read retry

This event occurs only if the instance is mounted in shared mode (Parallel Server). 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-11.

block# — See "block#" on page A-10.

checkpoint completed

A session waits for a checkpoint to complete. This could happen, for example, during a close database or a local checkpoint.

Wait Time: 5 seconds

Parameters: none

checkpoint range buffer not saved

During a range checkpoint operation a buffer was found that was not saved or written. Either:

- the session will wait on this event if the write batch is empty and it is the first time that the session waited on this event in the range checkpoint operation
- the current range checkpoint operation will be aborted and a new one will be started to complete the operation

Wait Time: 10 milliseconds

Parameters: none

control file parallel write

This event occurs while the session is writing physical blocks to all controlfiles. This happens when:

- the session starts a controlfile transaction (to make sure that the controlfiles are up to date in case the session crashes before committing the controlfile transaction)
- the session commits a transaction to a controlfile
- changing a generic entry in the controlfile, the new value is being written to all controlfiles

Wait Time: The wait time is the time it takes to finish all writes to all controlfiles.

Parameters:

files	The number of controlfiles to which the session is writing.
blocks	The number of blocks that the session is writing to the controlfile.

requests The number of I/O requests which the session wants to write.

control file sequential read

Reading from the controlfile. 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 controlfile'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 controlfile to which the session is currently writing.

block# Block number in the controlfile 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 controlfile to which the session is currently writing.

block# Block number in the controlfile 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-11.
block#	See "block#" on page A-11.
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 controlfile, 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-11.
block#	See"block#" on page A-11.

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-11.

block# See"block#" on page A-11.

blocks This is the number of blocks that the session is trying to write in *file#* starting at *block#*.

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, "Enqueue and Lock Names").

Wait Time: Depends on the enqueue name.

Parameters:

name	See "name and type" on page A-12.
mode	See "mode" on page A-11.

file identify

The time it takes to identify a file so that it can be opened later.

file open

The time it takes to open the file.

free buffer waits

This will happen if:

- All buffer gets have been suspended. This could happen when a file was read-only and is now read-write. All the existing buffers need to be invalidated since they are not linked to lock elements (needed when mounted parallel (shared)). So cache buffers are not assigned to data block addresses until the invalidation is finished.
- The session moved some dirty buffers to the dirty queue and now this dirty queue is full. The dirty queue needs to be written first. The session will wait on this event and try again to find a free buffer
- This also happens after inspecting **free buffer inspected** buffers. If no free buffer is found, Oracle waits for one second, and then tries to get the buffer again (depends on the context). For more information, see free buffer inspected on page C-9.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-11.
block#	See "block#" on page A-10.

free global transaction table entry

The session is waiting for a free slot in the global transaction table (used by the Distributed Database Option). It will wait for 1 second and try again.

Wait Time: 1 second

Parameter:

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-11.
block#	See"block#" on page A-11.
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-11.
block#	See"block#" on page A-11.
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-11.
block#	See"block#" on page A-11.
class	See "class" on page A-10.

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-11.
block#	See"block#" on page A-11.
lenum	See"lenum" on page A-11.

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-11.
block#	See"block#" on page A-11.
class	See"class" on page A-11.

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-11.
block#	See"block#" on page A-11.
class	See"class" on page A-11.

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-11.
block#	See"block#" on page A-11.
lenum	See"lenum" on page A-11.

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-11.
block#	See"block#" on page A-11.
lenum	See"lenum" on page A-11.

inactive session

This event is used for two purposes:

- Switching sessions

  If a time-out period has been specified, then wait that amount of time for the session to be detached.

- Killing sessions

  From either KILL SESSION or internal request. Having posted a session that it should kill itself, wait for up to 1 minute for the session to die.

Wait Time: 1 second

Parameters:

session#	See "session#" on page A-12.
waited	See "waited" on page A-12.

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-12.

index block split

While trying to find an index key in an index block, Oracle noticed that the index block was being split. Oracle will wait for the split to finish and try to find the key again.

Wait Time: The session will yield the CPU, so there is no actual waiting time.

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#;`

```
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 this SQL command: `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$LATCHNAME view.To find more information on the latch, use this SQL command:
	`select *`
	`from v$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 (e.g., 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-12.

library cache pin

This event manages library cache concurrency. Pinning an object causes the heaps to be loaded into memory. If a client wants to modify or examine the object, the client must acquire a pin after the lock.

Wait Time: 3 seconds (1 second for PMON)

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-12.

lock manager wait for remote message

The lock manager waits for a message from a remote lock manager in the same configuration.

Wait Time: The elapsed time of the wait

Parameter:

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-10.
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-10.
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 sync'ed

log switch/archive

Used as part of the ALTER SYSTEM ARCHIVE LOG CHANGE *scn* command. 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.

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 Multi-Threaded Server (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 Queue) 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, LCK0) 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-11.
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.

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.

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

Enqueue and Lock Names

This appendix describes the enqueues and locks that are used by Oracle. Locks and Note that resources are different structures used by Oracle, but sometimes the names are not used correctly. A *resource* uniquely identifies a certain object that can be locked by different sessions within an instance (Local Resource) or between instances (Global Resource). Each session will have a lock structure on the resource if it tries to lock the resource. Topics covered in this appendix include:

- Oracle Enqueue and Lock Names

Oracle Enqueue and Lock Names

Table B–1 contains a list of the enqueues and locks that are used by Oracle. .

Note: The names of enqueues and locks and their definitions may change from release to release.

Table B–1 Oracle Enqueue and Lock Names

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

Table B–1 Oracle Enqueue and Lock Names

WL, Being Written Redo Log	XA, Instance Attribute Lock
XI, Instance Registration Lock	------

C

Statistics Descriptions

This appendix briefly describes some of the statistics stored in the V$SESSTAT and V$SYSSTAT dynamic performance tables. These statistics are useful in identifying and correcting performance problems.

Topics covered in this appendix include:

Displaying Statistics

The V$SESSTAT view contains statistics on a per-session basis and is only valid for the session currently connected. When a session disconnects all statistics for the session are updated in V$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. This may even happen during a bug fix patch release. Application developers should be aware of this and write their code to tolerate missing or extra statistics.

For more information on these views, see "V$SESSTAT" on page 3-103, "V$STATNAME" on page 3-115, and "V$SYSSTAT" on page 3-119.

Statistics Descriptions

This section describes the statistics stored in the V$SESSTAT and V$SYSSTAT views.

CPU used by this session

This is the amount of CPU time (in 10s of milliseconds) used by a session between when a user call started and ended. Some user calls can complete within 10 milliseconds and as a result, the start and end user-call time can be the same. In this case, 0 milliseconds are added to the statistic.

A similar problem can exist in the reporting by the operating system, especially on systems that suffer from many context switches.

CPU used when call started

The CPU time used when the call is started.

CR blocks created

A buffer in the buffer cache was cloned. The most common reason for cloning is that the buffer is held in a incompatible mode.

Cached Commit SCN referenced

The number of times cached Commit SCN is referenced.

Commit SCN referenced

The number of times Commit SCN is cached.

Current blocks converted for CR

A CURRENT buffer (shared or exclusive) is made CR before it can be used.

DBWR buffers scanned

The total number of buffers looked at when scanning each LRU set for dirty buffers to clean. This count includes both dirty and clean buffers. Divide by **DBWR lru scans** to find the average number of buffers scanned.

DBWR checkpoint buffers written

The number of buffers that were written for checkpoints.

DBWR checkpoints

Number of times the DBWR was asked to scan the cache and write all blocks marked for a checkpoint.

DBWR forced writes

DFS only: count of the number of blocks forced written.

DBWR free buffers found

The number of buffers that DBWR found to be clean 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

The number of times that DBWR does a scan of the LRU queue looking for buffers to write. This includes times when the scan is 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

Number of messages received requesting DBWR to make some more free buffers for the LRU.

DBWR revisited being-written buffer

The number of times that dbwr tried to save a buffer for writing and found that it was already in the write batch. This statistic is a measure of the amount of "useless" work that DBWR had to do in trying to fill the batch. This can occur because 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" since the buffer is already marked as being written.

DBWR skip hot writes

The number of times DBWR skipped writing "hot" buffers.

DBWR summed scan depth

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 undo block writes

The number of transaction table blocks written by DBWR. It is an indication of how many "hot" buffers were written, leading to write complete waits.

DDL statements parallelized

The number of DDL statements that were parallelized.

DML statements parallelized.

The number of DML statements that were parallelized.

PX local messages recv'd

The number of local messages received for Parallel Executions.

PX local messages sent

The number of local messages send for Parallel Executions.

PX remote messages recv'd

The number of remote messages received for Parallel Executions.

PX remote messages sent

The number of remote messages sent for Parallel Executions.

SQL*Net roundtrips to/from client

Total number of Net8 messages sent to and received from the client.

SQL*Net roundtrips to/from dblink

Total number of Net8 messages sent over and received from a database link.

Switch current to new buffer

The number of times the current version moved to a different buffer, leaving CR.

Unnecessary process cleanup for SCN batching

The total number of times that the process cleanup was performed unnecessarily because the session / process did not get the next batched SCN. The next batched SCN went to another session instead.

background checkpoints completed

The number of checkpoints completed by the background. This statistic is incremented when the background successfully advances the thread checkpoint.

background checkpoints started

The number of checkpoints started by the background. It can be larger than the number completed if a new checkpoint overrides an incomplete checkpoint. This only includes checkpoints of the thread, not individual file checkpoints for operations such as offline or begin backup. This statistic does not include the checkpoints performed in the foreground, such as ALTER SYSTEM CHECKPOINT LOCAL.

bytes received via SQL*Net from client

The total number of bytes received from the client over Net8.

bytes received via SQL*Net from dblink

The total number of bytes received from a database link over Net8.

bytes sent via SQL*Net to client

The total number of bytes sent to the client from the foreground process(es).

bytes sent via SQL*Net to dblink

The total number of bytes sent over a database link.

calls to get snapshot scn: kcmgss

The number of times a snap System Change Number (SCN) was allocated. The SCN is allocated at the start of a transaction.

change write time

The elapsed time for redo write for changes made to CURRENT blocks in 10s of milliseconds.

cleanouts and rollbacks - consistent read gets

The number of times CR gets require both block rollbacks, and block cleanouts.

cleanouts only - consistent read gets

The number of times CR gets require only block cleanouts, no rollbacks.

cluster key scan block gets

The number of blocks obtained in a cluster scan.

cluster key scans

The number of cluster scans that were started.

commit cleanout failures: block lost

The number of times a cleanout at commit was attempted and could not find the correct block due to forced write, replacement, or switch CURRENT.

commit cleanout failures: buffer being written

The number of times a cleanout at commit was attempted but the buffer was currently being written.

commit cleanout failures: callback failure

The number of times the cleanout callback function returns FALSE.

commit cleanout failures: cannot pin

The 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

The number of times cleanout at commit was attempted during hot backup. The image of the block needs to be logged before the buffer can be made dirty.

commit cleanout failures: write disabled

The number of times that a cleanout at commit time was performed but the writes to the database had been temporarily disabled.

commit cleanouts

The total number of times the cleanout block at commit time function was performed.

commit cleanouts successfully completed

The number of times the cleanout block at commit time function successfully completed.

consistent changes

The number of times a database block 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.

consistent gets

The number of times a consistent read was requested for a block. See also consistent changes above.

data blocks consistent reads - undo records applied

The number of undo records applied to CR rollback data blocks.

db block changes

Closely related to **consistent changes**, this statistics counts the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These are changes that are generating redo log entries and hence will be permanent changes to the database if the transaction is committed.

This statistic is a rough indication of total database work. This statistic indicates (possibly on a per-transaction level) the rate at which buffers are being dirtied.

db block gets

This statistic tracks the number of blocks obtained in CURRENT mode.

deferred (CURRENT) block cleanout applications

The number of times cleanout records are deferred, piggyback with changes, always current get.

dirty buffers inspected

The number of dirty buffers found by the foreground while the foreground is looking for a buffer to reuse.

enqueue conversions

The total number of enqueue converts.

enqueue deadlocks

The total number of enqueue deadlocks between different sessions.

enqueue releases

The total number of enqueue releases.

enqueue requests

The total number of enqueue gets.

enqueue timeouts

The total number of enqueue operations (get and convert) that timed out before they could complete.

enqueue waits

The total number of waits that happened during an enqueue convert or get because the enqueue could not be granted right away.

exchange deadlocks

The number of times that a process detected a potential deadlock when exchanging two buffers and raised an internal, restartable error. Index scans are currently the only operations which perform exchanges.

execute count

The total number of calls (user and recursive) that execute SQL statements.

free buffer inspected

The 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 were busy, needed to be written after rapid aging out, or they have a user, a waiter, or are being read / written. For more information, see "dirty buffers inspected".

free buffer requested

The count of the number of times a reusable buffer or a free buffer was requested to create or load a block.

global cache defers

The number of times a ping request was deferred until later.

global cache freelist waits

The number of pings for free lock elements (when all releasable locks are in use)

global lock convert time

The total elapsed time of all synchronous (non-asynchronous) global lock converts in 10s of milliseconds.

global lock converts (async)

The total number of asynchronous global lock converts.

global lock converts (non async)

The total number of synchronous global lock converts.

global lock get time

The total elapsed time of all synchronous (non-asynchronous) global lock gets in 10s of milliseconds.

global lock gets (async)

The total number of asynchronous global lock gets.

global lock gets (non async)

The total number of synchronous global lock gets.

global lock releases

The total number of synchronous global lock releases.

hot buffers moved to head of LRU

When a hot buffer reaches the tail of its replacement list, it is moved back to the head of the list. This is what keeps hot buffers from being reused. This statistic indicates how often that happens.

immediate (CR) block cleanout applications

The number of times cleanout records are applied immediately during CR gets.

immediate (CURRENT) block cleanout applications

The number of times cleanout records are applied immediately during current gets.

kcmccs called get current scn

The number of times the kernel got the CURRENT SCN when there was a need to casually confirm the SCN.

kcmccs read scn without going to DLM

The number of times the kernel casually confirmed the SCN without going to the LM.

kcmgss waited for batching

The number of times the kernel waited on a snapshot SCN.

logons cumulative

The total number of logons since the instance started. This statistic is useful only in V$SYSSTAT. It gives an instance overview of all processes that logged on.

logons current

The total number of current logons. This statistic is useful only in V$SYSSTAT.

native hash arithmetic execute

Incremented when the native arithmetic runtime engine is invoked.

native hash arithmetic fail

Incremented when the runtime engine encounters an overflow condition.

next scns gotten without going to DLM

The number of SCNs (System Change Numbers) obtained without going to the DLM.

no work - consistent read gets

The number of times CR gets require no block cleanouts nor rollbacks.

opened cursors cumulative

The total number of opened cursors since the instance has started (in V$SYSSTAT). In V$SESSTAT, this statistic shows the total number of cursors opened since the start of the session.

opened cursors current

The total number of current open cursors.

opens of replaced files

The total number of files that needed to be reopened because they were no longer in the process file cache.

opens requiring cache replacement

The total number of file opens that caused a current file to be closed in the process file cache.

parse count (hard)

The total number of parse calls (real parses). A hard parse means allocating a workheap and other memory structures, and then building a parse tree. A hard parse is a very expensive operation in terms of memory use.

parse count (total)

Total number of parse calls (hard and soft). A soft parse is a check to make sure that the permissions on the underlying object have not changed.

parse time cpu

The total CPU time used for parsing (hard and soft) in 10s of milliseconds.

parse time elapsed

The total elapsed time for parsing in 10s of milliseconds. By subtracting **parse time cpu** from the this statistic, the total waiting time for parse resources is determined. For more information see parse time cpu above.

physical reads

The total number of data blocks read from disk. This equals the number of "physical reads direct" plus all reads into buffer cache.

physical reads direct

The number of reads directly read 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

The total number of data blocks written to disk. This equals the number of "physical writes direct" plus all writes from buffer cache.

physical writes direct

The number of writes directly written to disk bypassing the buffer cache (as in a direct load operation).

physical writes non-checkpoint

The number of writes that would occur were checkpointing turned off. Note that this is a theoretical number because checkpointing will always be required for log switches.

pinned buffers inspected

The number of times a foreground encountered a cold buffer that was pinned or had a waiter that was about to pin it when the foreground is scanning the tail of the replacement list looking for a buffer to reuse. It should be uncommon because a cold buffer should not be pinned very often.

queries parallelized

The number of SELECT statements that got parallelized.

recovery array read time

The elapsed time of I/O while doing recovery.

recovery array reads

The number of reads performed during recovery.

recovery blocks read

The number of blocks read during recovery.

recursive calls

Oracle maintains tables used for internal processing. When Oracle needs to make a change to these tables, it internally generates a SQL statement. These internal SQL statements generate recursive calls.

recursive cpu usage

The 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 entries

This statistic increments each time redo entries are copied into the redo log buffer.

redo log space requests

The active log file is full and Oracle is waiting for disk space to be allocated for the redo log entries. Space is created by performing a log switch.

Small Log files 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

The total elapsed time of waiting for **redo log space request** in 10s of milliseconds.

redo log switch interrupts

The number of times that another instance asked this instance to advance to the next log file.

redo ordering marks

The number of times that an SCN had to be allocated to force a redo record to have an higher SCN than a record generated in another thread using the same block.

redo size

The total amount of redo generated in bytes.

redo synch time

The elapsed time of all **redo sync writes** calls in 10s of milliseconds.

redo sync writes

Usually, redo that is generated and copied into the log buffer need not be flushed out to disk immediately. The log buffer is a circular buffer that LGWR periodically flushes. Redo sync writes increments when changes being applied must be written out to disk due to a commit.

redo wastage

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

The total elapsed time of the write from the redo log buffer to the current redo log file in 10s of milliseconds.

redo writer latching time

The elapsed time need by LWGR to obtain and release each copy latch in 10s of milliseconds. This is only used if the initialization parameter LOG_SIMULTANEOUS_COPIES > 0. For more information, see "LOG_SIMULTANEOUS_COPIES" on page 1-70.

redo writes

Count of the total number of writes by LGWR to the redo log files.

remote instance undo block writes

The number of times this instance wrote a dirty undo block so that another instance could read it.

remote instance undo header writes

The number of times this instance wrote a dirty undo header block so that another instance could read it.

rollback changes - undo records applied

The number of undo records applied to rollback (real) changes.

rollbacks only - consistent read gets

The number of times CR gets require only block rollbacks, no block cleanouts.

serializable aborts

The number of times a SQL statement in serializable isolation level had to abort.

session connect time

The connect time for the session in 1/100 seconds. This value is useful only in V$SESSTAT. It is the wall clock time of when the logon to this session occurred.

session cursor cache count

The total number of cursor cached. This is only incremented 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 initialization parameter SESSION_CACHED_CURSORS, the value of the initialization parameter should be increased.

session cursor cache hits

The count of the number of hits in the session cursor cache. A hit means that the SQL statement did not have to be reparsed. By subtracting this statistic from **parse count (total)** one can determine the real number of parses that happened.

session logical reads

This statistic is basically **db block gets** + **consistent gets**. For more information, see "db block gets" on page C-8 and "consistent gets" on page 8.

session pga memory

This statistic shows the current PGA size for a session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.

session pga memory max

This statistic shows the peak PGA size for a session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.

session stored procedure space

This statistic shows the amount of memory that this session is using for stored procedures.

session uga memory

This statistic shows the current UGA size for a session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.

session uga memory max

This statistic shows the peak UGA size for a session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.

sorts (disk)

If the number of disk writes is non-zero for a given sort operation, then this statistic is incremented.

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 1-118.

sorts (memory)

If the number of disk writes is zero, then the sort was performed completely in memory and this statistic is incremented.

This is more an indication of sorting activity in the application work load. 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)

The total number of rows sorted.

summed dirty queue length

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.

table fetch by rowid

When rows are fetched using a ROWID (usually recovered from an index), each row returned increments this counter.

This statistic is an indication of row fetch operations being performed with the aid of an index. Because doing table scans usually indicates either non-optimal queries or tables without indexes, this statistic should increase as the above issues have been addressed in the application.

table fetch continued row

When a row that spans more than one block is encountered during a fetch, this statistic is incremented.

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. Taking a closer look at the STORAGE parameters PCT_FREE and PCT_USED. 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 scan blocks gotten

During scanning operations, each row is retrieved sequentially by Oracle. Each block encountered during the scan increments this statistic.

This statistic informs you of the number of database blocks that you had to get from the buffer cache for the purpose of scanning. Compare the value of this parameter to the value of **consistent gets** to get a feeling for how much of the consistent read activity can be attributed to scanning. For more information, see "consistent gets" on page C-8.

table scan rows gotten

This statistic is collected during a scan operation, but instead of counting the number of database blocks, it counts the rows being processed.

table scans (cache partitions)

Count of range scans on tables that have the CACHE option enabled.

table scans (direct read)

Count of table scans performed with direct read (bypassing the buffer cache).

table scans (long tables)

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) below.

table scans (rowid ranges)

Count of table scans with specified ROWID endpoints. This is performed for Parallel Query.

table scans (short tables)

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

The total number of file opens being 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 rollbacks

The number of transactions being successfully rolled back.

transaction tables consistent read rollbacks

The number of times transaction tables are CR rolled back.

transaction tables consistent reads - undo records applied

The number of undo records applied to CR rollback transaction tables.

user calls

Oracle allocates resources (Call State Objects) to keep track of relevant user call data structures every time you log in, parse, 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

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

This statistic stores the number of times users manually issue the ROLLBACK statement or an error occurs during users' transactions.

Index