Oracle Server 9*i*
Quick Reference Guide

Disclaimer

This document is nothing official by Oracle Corporation. Oracle holds all rights on its documentation. Certain terms are registered trademarks.

This quick reference guide is some kind of outgrown cheat-sheet for all of us with limited supply of system memory. It enables quick lookup of syntax for statements which one might need less frequently in daily work. So the major goal of this document is compaction, not clarity. Some syntax options only work under certain circumstances, some options exclude each other. This behaviour is intentionally not reflected to avoid proliferation of similar statements. Be aware of your actions! The author disclaims liability for errors within this document and subsequent mistakes that might harm your database. In case of uncertainties please refer to Oracle's excellent original documentation, which can be found online at the Oracle Technology Network (otn.oracle.com) – comprising several hundred megabytes meanwhile – or contact Oracle Customer Support. In any quick reference guide there is also no room for discussing concepts and techniques. If you do not know where to start just read the Oracle Database Concepts Guide which is very concise. Advanced discussions can be found elsewhere, my favourite resource being asktom.oracle.com. Oracle Guru Thomas Kyte has also written excellent books on techniques and best practices. Other invaluable repositories are the sites of Jonathan Lewis (www.jlcomp.demon.co.uk) and of Steve Adams (www.ixora.com.au). Not to forget that lots of technical whitepapers can be found at Oracle's Technet.

This reference uses a modified Backus-Naur Form syntax which is adapted from the Oracle online documentation. Optional parts are enclosed in square brackets [], a list of items is enclosed in curly braces {}, alternatives within brackets or braces are separated by a vertical bar |. Keywords appear in regular style and are not case-sensitive in most OS. Placeholders for input data are enclosed in brackets <> with corresponding default values underlined. A comma followed by an ellipsis inclosed in square brackets [, …] indicates that the preceding syntactic element may be repeated. An ellipsis without preceding comma … indicates that the corresponding syntactic elements have been specified beforehand.

Each section commonly starts with dynamic performance views and static data dictionary views (only DBA variants listed) and – occasionally – tables. Initialization parameters follow as well as database packages and important files or processes. Then comes a list of performance tuning measures and desupported or deprecated features. The main part of each section is comprised of SQL statements grouped by functionality. The final section may state utilities if any exist.

Changes from Releases 8*i* to 9*i* are colored blue for new features or red for obsolete features. Some features of Server9*i* Release 2 have been added without another special color highlighting than blue but I did not scan the whole Release 2 documentation yet. And probably it will not take too long until Server10*i* comes out…
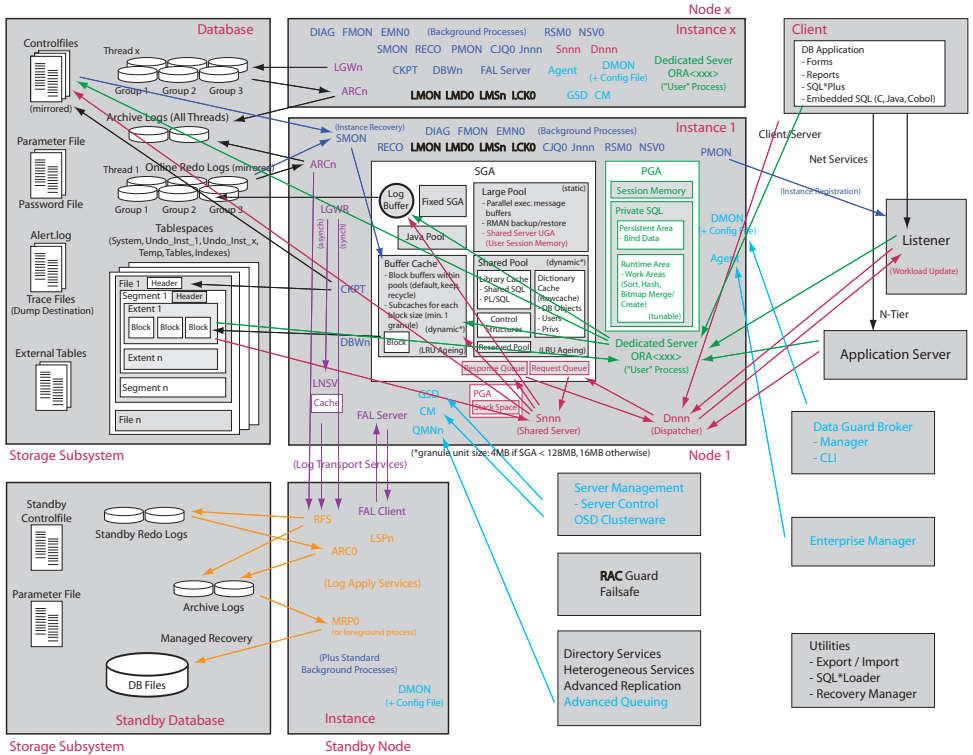
Outdated syntax has not been removed most of the time as one eventually comes across databases running older releases of Oracle Server. – One of my customers still uses Server7.

Any suggestions and corrections to improve this guide are welcome.

Oracle Server Architecture

Database

Controlfiles
(mirrored)

Thread x
Group 1 Group 2 Group 3

Archive Logs (All Threads)

Parameter File

Password File

Alert.log

Trace Files
(Dump Destination)

External Tables

Thread 1 Online Redo Logs (mirrored)
Group 1 Group 2 Group 3

Tablespaces
(System, Undo_Inst_1, Undo_Inst_x,
Temp, Tables, Indexes)

File 1 Header
Segment 1 Header
Extent 1
Block Block Block Block
Extent n
Segment n
File n

Storage Subsystem

Node x

Instance x

DIAG FMON EMN0 (Background Processes) RSM0 NSV0
SMON RECO PMON CJQ0 Jnnn Snnn Dnnn
CKPT DBWn FAL Server Agent DMON (+ Config File)
LMON LMD0 LMSn LCK0 GSD CM

Dedicated Sever
ORA<xxx>
("User" Process)

LGWn

ARCn

Instance 1

(Instance Recovery) SMON

DIAG FMON EMN0 (Background Processes)
RECO **LMON LMD0 LMSn LCK0** CJQ0 Jnnn RSM0 NSV0

ARCn

LGWR

CKPT

DBWn

LNSV
Cache

FAL Server

GSD
CM
QMNn

PGA
Stack Space

PMON

SGA

Log Buffer | Fixed SGA | Large Pool (static)
- Parallel exec. message buffers
- RMAN backup/restore
- Shared Server UGA (User Session Memory)

Java Pool

Buffer Cache
- Block buffers within pools (default, keep, recycle)
- Subcaches for each block size (min. 1 granule) (dynamic*)
Block (LRU Ageing)

Shared Pool (dynamic*)
Library Cache
- Shared SQL
- PL/SQL
Control Structures
Reserved Pool (LRU Ageing)

Dictionary Cache (rowcache)
- DB Objects
- Users
- Privs

Response Queue Request Queue

PGA
Session Memory
Private SQL
Persistent Area
- Bind Data
Runtime Area
- Work Areas (Sort, Hash, Bitmap Merge/ Create) (tunable)

Dedicated Server
ORA<xxx>
("User" Process)

Snnn
(Shared Server)

Dnnn
(Dispatcher)

Node 1

(*granule unit size: 4MB if SGA < 128MB, 16MB otherwise)

DMON (+ Config File)

Agent

Client

DB Application
- Forms
- Reports
- SQL*Plus
- Embedded SQL (C, Java, Cobol)

Client/Server

Net Services

(Instance Registration)

Listener
(Workload Update)

N-Tier

Application Server

Data Guard Broker
- Manager
- CLI

Enterprise Manager

Server Management
- Server Control
OSD Clusterware

RAC Guard
Failsafe

Directory Services
Heterogeneous Services
Advanced Replication
Advanced Queuing

Utilities
- Export / Import
- SQL*Loader
- Recovery Manager

Standby Database

Standby Controlfile

Standby Redo Logs

Parameter File

Archive Logs

Managed Recovery

DB Files

Storage Subsystem

Standby Node

(Log Transport Services)

RFS

FAL Client

LSPn

ARC0

(Log Apply Services)

MRP0
(or foreground process)

(Plus Standard Background Processes)

DMON (+ Config File)

Instance

Instance

Background Processes (v$bgprocess)

ARC<n>, CJQ0, J<nnn>, CKPT, DBW<n>, DIAG, DMON, EMN0, LCK<n>*, LGWR, LMD0*, LMON*, LMS<n>*, LNSV, LSP0, MRP0, NSV0, PMON, QMN<n>, RECO, RLAP, RSM0, SMON, RFS<n>

BSP<n>*, SNP<n> << obsolete
* RAC processes

Failure of LGWR (Err 470), CKPT (470), DBW<n> (471), ARC<n> (473), SMON (474) or RECO (476) lead to termination of instance by PMON. Failure of PMON leads to termination of instance by DBW<n> (Err 472).

Failed SNP<n> processes are restarted by PMON.

Foreground Processes

D<nnn>, S<nnn>, P<nnn>

Views & Tables

v$fixed_table, v$fixed_view_definition, v$indexed_fixed_column, v$instance, vsga, vsgastat, v$pgastat, v$session, v$process, v$bgprocess, v$version, product_component_version, v$license, v$option, v$access, v$timer, v$parameter, v$parameter2, v$spparameter, v$system_parameter, v$system_parameter2, v$obsolete_parameter, vsql, vsqlarea, v$sqltext, v$sqltext_with_newlines, vsql_cursor, vsql_bind_data, v$sql_bind_metadata, v$sql_shared_memory, vsql_plan, vsql_workarea, v$sql_workarea_active, v$librarycache, v$rowcache, v$rowcache_parent, v$rowcache_subordinate, v$open_cursor, v$object_dependency, v$db_object_cache, v$shared_pool_reserved, v$bh, xbh, vcache, v$subcache, v$buffer_pool, v$buffer_pool_statistics, vdb_cache_advice, vstatistics_level, v$filestat, v$tempstat, v$sysstat, v$sesstat, v$mystat, v$statname, v$waitstat, v$latch, v$latchname, v$latchholder, v$latch_parent, v$latch_children, v$event_name, v$system_event, v$session_event, v$session_wait, v$sess_io, v$segment_statistics, v$segstat, v$segstat_name, v$circuit, v$queue, v$shared_server, v$shared_server_monitor, v$dispatcher, v$dispatcher_rate, v$reqdist, v$queue, v$lock, v$enqueue_lock, v$enqueue_stat, v$locked_object, v$global_blocked_locks, dba_locks, dba_lock, dba_lock_internal, v$session_connect_info, v$session_longops, v$system_cursor_cache, v$session_cursor_cache, v$session_object_cache, v$bsp, v$px_session, v$px_sesstat, v$px_process, v$px_process_sysstat, v$pq_sesstat, v$pq_slave, v$pq_sysstat, v$pq_tqstat, v$execution, v$mls_parameters, deptree, session_context

Parameters (init<sid>.ora)

spfile, ifile, instance_name, service_names, db_block_size, sga_max_size, db_cache_size, db_keep_cache_size, db_recycle_cache_size, db_<n>k_cache_size, db_cache_advice, shared_pool_size, log_buffer, large_pool_size, java_pool_size, shared_pool_reserved_size, pre_page_sga, sessions, processes, user_dump_dest, background_dump_dest, max_dump_file_size, local_listener, remote_listener, mts_service, circuits, dispatchers, max_dispatchers, shared_servers, max_shared_servers, shared_server, shared_server_sessions, dbwr_io_slaves, remote_os_authent, os_authent_prefix, dml_locks, enqueue_resources, parallel_automatic_tuning, parallel_min_servers, parallel_max_servers, parallel_min_percent, parallel_adaptive_multi_user, parallel_threads_per_cpu, parallel_execution_message_size, parallel_broadcast_enabled, oracle_trace_enable, oracle_trace_collection_{name | path | size}, oracle_trace_facility_{name | path}, java_soft_sessionspace_limit, java_max_sessionspace_size, lock_sga, shared_memory_address, hi_shared_memory_address, object_cache_optimal_size, object_cache_max_size_percent, serial_reuse, session_max_open_files, timed_os_statistics, cursor_sharing, drs_start

Packages

DBMS_SYSTEM
    set_sql_trace_in_session
DBMS_SUPPORT
    mysid, {start | stop}_trace, {start | stop}_trace_in_session
DBMS_SESSION
    set_sql_trace, {set | clear}_identifier, {set | list | clear}_context, set_role, set_nls, is_role_enabled, is_session_alive, unique_session_id, close_database_link, reset_package, modify_package_state, switch_current_consumer_group, free_unused_user_memory, set_close_cached_open_cursors
DBMS_SHARED_POOL
    keep, unkeep, sizes
DBMS_APPLICATION_INFO
    set_module, set_action, set_client_info, read_module, read_client_info

Files

dbmspool.sql, dbmssupp.sql, catparr.sql, utldtree.sql

Tuning/Contention

Statistics classes:
    1 User, 2 Redo, 4 Enqueue, 8 Cache, 16 OS, 32 RAC, 64 SQL, 128 Debug

Buffer cache: «Cache Hit Ratio» (v$sysstat) or per pool (v$buffer_pool_statistics)
    1 – («physical reads» / («db block gets» + «consistent gets»)) < 90–95% -> increase «db_block_buffers» or «buffer_pool_keep», «buffer_pool_recycle»

Shared pool: «Shar. Cursors» (v$librarycache) gethitratio for SQL AREA < 99%

Library cache: sum(reloads) / sum(pins) > 1% (v$librarycache)

Dict. cache: sum(getmisses) / sum(gets) > 15% (v$rowcache) -> increase «shared_pool_size»

LRU latch: «cache buffers lru chain» (v$latch) misses / gets > 1% -> increase «db_block_lru_latches» (max. CPU * 2 or BUFFERS / 50)

Redo buffer: «redo%retries» (v$sysstat)

PGA: «%ga memory%» (v$sysstat), «sorts%» (v$sysstat), sorts (v$sqlarea), «workarea%» (v$sysstat, v$sesstat), v$pgastat, v$sql_workarea, v$sql_workarea_active, pga_%_mem (v$process)

Deprecated Features

v$mts

db_block_buffers, buffer_pool_keep, buffer_pool_recycle, mts_circuits, mts_dispatchers, mts_max_dispatchers, mts_servers, mts_sessions

utlbstat.sql, utlestat.sql

Desupported Features

v$recent_bucket, v$current_bucket, db_block_lru_latches, use_indirect_data_buffers, db_block_lru_extended_statistics, db_block_lru_statistics, lock_sga_areas, shared_pool_reserved_min_alloc, parallel_server_idle_time, parallel_transaction_resource_timeout, parallel_min_message_pool, mts_rate_log_size, mts_rate_scale, mts_max_servers

Instance (cont.)

Parameters

show parameter[s] <string>

alter system set <param> [=] <value>
    [comment '<text>'] [deferred]
    [scope = {memory | spfile | both}]
    [sid = { '<sid>' | '*' }];

alter system reset <param>
    [scope = {memory | spfile | both}]
    [sid = { '<sid>' | '*' }];

Static Initialization Parameters

active_instance_count = <n>, audit_file_dest = <dir>, audit_trail = {none | false | db | true | os}, background_core_dump = {full | partial}, bitmap_merge_area_size = <1m>, blank_trimming = {true | false}, buffer_pool_{keep | recycle} = {<n> | (buffers: <n>, lru_latches: <n>} << deprecated, circuits = <n>, cluster_database = {true | false}, cluster_database_instances = <n>, cluster_interconnects = <ip>[:<ip>…], commit_point_strength = <n>, compatible = <x.x.x>, control_files = ("<file>" [, …]), cpu_count = <n>, create_bitmap_area_size = <8m>, cursor_space_for_time = {true | false}, db_block_buffers = <n> << deprecated, db_block_size = <2048>, db_domain = <str>, {db | log}_file_name_convert = ('prim', 'stdby' [, …]), db_files = <200>, db_name = <str>, db_writer_processes = <1>, dblink_encrypt_login = {true | false}, dbwr_io_slaves = <0>, disk_asynch_io = {true | false}, distributed_transactions = <n>, gc_files_to_locks = '<f#>[-<f#>]=<n>[!<B>][r][each][: …]' << disables Cache Fusion, hi_shared_memory_address = <0>, ifile = <file>, instance_groups = <gr> [, …], instance_name = <sid>, instance_number = <n>, java_max_sessionspace_size = <0>, java_pool_size = <20k>, java_soft_sessionspace_limit = <0>, large_pool_size = <n>, local_listener = <serv>, lock_name_space = <name>, lock_sga = {true | false}, log_archive_format = <fmt>, log_archive_start = {true | false}, log_buffer = <n>, logmnr_max_persistent_sessions = <1>, max_commit_propagation_delay = <700>, max_dispatchers = <5>, max_enabled_roles = <20>, max_shared_servers = <n>, o7_dictionary_accessibility = {true | false}, open_cursors = <50>, open_links = <4>, open_links_per_instance = <4>, optimizer_features_enable = <9.0.0>, oracle_trace_collection_name = <name>, oracle_trace_collection_path = <dir>, oracle_trace_collection_size = <n>, oracle_trace_facility_name = {oracled, oraclee, oraclesm, oraclec}, oracle_trace_facility_path = <dir>, os_authent_prefix = <OPS$>, os_roles = {true, false}, parallel_automatic_tuning = {true | false}, parallel_execution_message_size = <n>, parallel_max_servers = <n>, parallel_min_servers = <0>, pre_page_sga = {true | false}, processes = <n>, rdbms_server_dn = <x.500>, read_only_open_delayed = {true | false}, recovery_parallelism = <n>, remote_archive_enable = {true | false}, remote_listener = <serv>, remote_login_passwordfile = {none | shared | exclusive}, remote_os_authent = {true

| false}, remote_os_roles = {true | false}, replication_dependency_tracking = {true | false}, rollback_segments = (<rbs> [, …]), row_locking = {always | default | intent}, serial_reuse = {disable | select | sml | plsql | all}, session_max_open_files = <10>, sessions = <(1.1*proc)+5>, sga_max_size = <n>, shadow_core_dump = {partial | full}, shared_memory_address = <0>, shared_pool_reserved_size = <5%SP>, shared_server_sessions = <n>, spfile = <file>, sql92_security = {true | false}, sql_trace = {true | false}, tape_asynch_io = {true | false}, thread = <n>, transactions_per_rollback_segment = <5>, undo_management = {manual | auto}, util_file_dir = <dir>

Dynamic Initialization Parameters

aq_tm_processes = <n>, archive_lag_target = <n>, background_dump_dest = <dir>, backup_tape_io_slaves = {true | false}, control_file_record_keep_time = <7>, core_dump_dest = '<dir>', db_{2|4|8|16|32}k_cache_size = <0>, db_block_checking = {true | false}, db_block_checksum = {true | false}, db_cache_advice = {on | ready | off}, db_cache_size = <48m>, db_{keep | recycle}_cache_size = <0m>, dispatchers = '{ (protocol = <prot>) | (description = (address =…)) | (address = (protocol = <prot>) (host = <node>) (port = <port>))} (connections = <n>) (dispatchers = <1>) (index = <n>) (listener = <list>) ({pool | multiplex} = {1 | on | yes | true | both | ({in | out} = <n>) | 0 | off | no | false | <n>}) (ticks = <15>) (service = <serv>) (presentation = {ttc | oracle.aurora.server.{SGiopServer | GiopServer} })', drs_start = {true | false}, fal_client = <serv>, fal_server = <serv>, fast_start_io_target = <n> << deprecated, fast_start_mttr_target = <0>, fast_start_parallel_rollback = {hi | lo | false}, fixed_date = <date>, global_context_pool_size = <1m>, hs_autoregister = {true | false}, job_queue_processes = <0>, license_max_sessions = <0>, license_max_users = <0>, license_sessions_warning = <0>, log_archive_dest = <dir>, log_archive_duplex_dest = <dir>, log_archive_max_processes = <1>, log_archive_trace = <0>, log_checkpoint_interval = <0>, log_checkpoint_timeout = <sec>, log_checkpoints_to_alert = {true | false}, parallel_adaptive_multi_user = {true | false}, parallel_threads_per_cpu = <n>, pga_aggregate_target = <0>, plsql_native_c_compiler = <path>, plsql_native_library_dir = <dir>, plsql_native_library_subdir_count = <0>, plsql_native_linker = <path>, plsql_native_make_file_name = <path>, plsql_native_make_utility = <path>, resource_limit = {true | false}, resource_manager_plan = <plan>, service_names = <serv> [, …], shared_pool_size = <16/64m>, shared_servers = <0/1>, standby_archive_dest = <path>, standby_file_management = {manual | auto}, trace_enabled = {true | false}, transaction_auditing = {true | false}, undo_retention = <900>, undo_tablespace = <ts>, user_dump_dest = <dir>

Session Scope Dynamic Init. Parameters

alter session set <param> [=] <value>;
cursor_sharing = {similar | exact | force}, db_block_checking, db_create_file_dest = '<dir>', db_create_online_log_dest_<1-5> = '<dir>', db_file_multiblock_read_count = <8>, global_names = {true | false}, hash_area_size = <n>, hash_join_enabled = {true | false}, log_archive_dest_<1-10> = {location = <path> | service = <serv>} [optional | mandatory] [[no]reopen [=<300>]] [arch | lgwr] [synch | async = <n>] [[no]affirm] [[no]delay [= <30>]] [[no]dependency] [[no]alternate [= <dest>]] [[no]max_failure [= <n>] [[no]quota_size [= <n>] [[no]quota_used] [[no]register = <loc>]], log_archive_dest_state_<1-10> = {enable | defer | alternate}, log_archive_min_succeed_dest = <1>, max_dump_file_size = {<n> | unlimited}, nls_calendar = '<cal>', nls_comp = {binary | ansi}, nls_currency = <curr>, nls_date_format = '<fmt>', nls_date_language = <lang>, nls_dual_currency = <curr>, nls_iso_currency = <terr>, nls_language = <lang>, nls_length_semantics = {byte | char}, nls_nchar_conv_excp = {true | false}, nls_numeric_characters = <sep>, nls_sort = {binary | <ling>}, nls_territory = <terr>, nls_time_format = '<fmt>', nls_timestamp_format = '<fmt>', nls_timestamp_tz_format = '<fmt>', nls_time_tz_format = '<fmt>', object_cache_max_size_percent = <10>, object_cache_optimal_size = <0>, optimizer_index_caching = <0>, optimizer_index_cost_adj = <100>, optimizer_max_permutations = <80000>, optimizer_mode = {first_rows_[1|10|100|1000] | first_rows | all_rows | choose | rule}, oracle_trace_enable = {true | false}, parallel_broadcast_enabled = {true | false}, parallel_instance_group = <gr>, parallel_min_percent = <0>, partition_view_enabled = {true | false}, plsql_compiler_flags = {[debug | non_debug], [interpreted | normal]}, plsql_v2_compatibility = {true | false}, query_rewrite_enabled = {true | false}, query_rewrite_integrity = {stale_tolerated | trusted | enforced}, remote_dependencies_mode = {timestamp | signature}, session_cached_cursors = <0>, sort_area_retained_size = <n>, sort_area_size = <65536>, star_transformation_enabled = {temp_disable | true | false}, statistics_level = {typical | basic | all}, timed_os_statistics = <0>, timed_statistics = {true | false}, tracefile_identifier = '<id>', undo_suppress_errors = {true | false}, workarea_size_policy = {auto | manual}

Session Parameters Only

constraint[s] = {immediate | deferred | default}, create_stored_outlines = {true | false | '<cat>'} [nooverride], current_schema = <schema>, error_on_overlap_time = {true | false}, flagger = {entry | immediate | full | off}, instance = <n>, isolation_level = {serializable | read committed}, plsql_debug = {true | false}, skip_unusable_indexes = {true | false}, sql_trace = {true | false}, time_zone = {'<[+|-]hh:mi>' | local | dbtimezone | '<tz_region>'}, use_{private |

Instance (cont.)

stored}_outlines = {true | <u>false</u> | '<cat>'}

Hidden Initialization Parameters

_system_trig_enabled, _log_simultane-
ous_copies, _log_io_size

Deprecated Initialization Parameters

mts_dispatchers, mts_servers
%_area%_size <<for dedicated server configurations

Obsolete Initialization Parameters

job_queue_interval, db_block_max_dirty_tar-
get, hash_multiblock_io_count = <n>

Events

{ alter system set event =
| alter session set events [=] }
    '<dbg_evt> trace name context
    {forever, level <n> | off}'
alter session set events [=]
    { 'immediate trace name
        { heapdump | blockdump | treedump
        | controlf | systemstate | buffers } level
<n>'
    | '<oerr> trace name errorstack level <n>
        [; name processstate level <n>]' }

Debug events
10015 (rollback), 10046 (process), 10049,
10051, 10053, 10210, 10211, 10212, 10231,
10232, 10235, 10248 (dispatcher), 10249
(shared server + dispatcher), 10257 (pmon), 10262,
10289 (hex dump), 10297 (oid caching), 10325
(control), 10408 (block keywords), 10520 (avoid
invalidations), 10619 (compatibility), 19027 (ctxx-
path), 29700 (v$ges_convert% views), 30441

oradebug
    { help [cmd]
    | setmypid
    | setospid <ospid>
    | setorapid <orapid> ['force']
    | dump <dump> <lev> [addr]
    | dumpsga [bytes]
    | dumplist
    | event <evt>
    | session_event <evt>
    | dumpvar {p | s | uga} <var> [lev]
    | setvar {p | s | uga} <var> <val>
    | peek <addr> <len> [lev]
    | poke <addr> <len> <val>
    | wakeup <orapid>
    | suspend
    | resume
    | flush
    | close_trace

    | tracefile_name
    | lkdebug
    | nsdbx
    | -G {<inst> | def | all}
    | -R {<inst> | def | all}
    | setinst {"<inst> [, …]" | all}
    | sgatofile <"path">
    | dmpcowsga <"path">
    | mapcowsga <"path">
    | hanganalyze [level]
    | ffbegin
    | ffderegister
    | ffterminst
    | ffresumeinst
    | ffstatus
    | core
    | ipc
    | unlimit
    | procstat
    | call <func> [<arg> , …] }

Instance Startup/Shutdown

startup
    [force] [restrict] [pfile=<par>] [nomount |
    [<u>exclusive</u> | parallel [retry] | shared [retry]]
    { mount [<db>] | <u>open</u>
    [read {only | write [recover]} | recover]
    [<db>] }]
shutdown
    [<u>normal</u> | transactional [<u>local</u>]
    | immediate | abort]
alter database [<db>]
    { mount [{standby | clone} database]
        [<u>exclusive</u> | parallel] << obsolete
    | dismount
    | open [read only | [<u>read write</u>]
        [resetlogs | <u>noresetlogs</u>]]
    | close [normal | immediate] };

Instance Modification

alter system {enable | disable} restricted
    session;
alter system {quiesce restricted | unquiesce};
alter system {suspend | resume};
alter system kill session '<SID>,<Serial#>'
    [immediate];
alter system disconnect session
    '<SID>,<Serial#>'
    [post_transaction | immediate];
alter system shutdown [immediate] 'D<nnn>';
alter system register;
alter system flush shared_pool;

Utilities

orapwd
    file=<file> password=<pwd> entries=<n>
oradim
    –{new | edit | delete | startup | shutdown}
    –{sid <SID> | srvc <serv>} –newsid <SID>
    –usrpwd <pwd> –intpwd <pwd>
    –maxusers <n> –startmode {a | m}
    –shutmode {a | i | n}
    –{starttype | shuttype}
        {srvc | inst | srvc, inst}
    –pfile <par> –timeout <n>
tkprof <trc> <out>
    [explain=<user>/<pwd>@<netserv>]
    [table=<tab>] [print=<n>] [sys=no]
    [insert=<file>] [record=<file>]
    [aggregate=<n>] [sort=<opt>]
otrcfmt
oemctl
    { {start | stop | status | ping}
        oms [<user>/<pwd>] | {start | stop}
    paging [<host> <name>] | {enable | dis-
    able | dump | export | import}
        eventhandler [<file>]
    | {import | export} registry [<file>]
        <user>/<pwd>@<repalias>
    | configure rws }
oemapp {console | dataguard}
vppcntl –start
vtm

Instance (cont.)

Database Locks (v$lock)

modes
  0 - none, 1 - null (NULL), 2 - row share (SS), 3 - row exclusive (SX), 4 - share (S), 5 - share row exclusive (SSX), 6 - exclusive (X)

user types and names
  TM dml enqueue, TX transaction enqueue, UL user-defined lock

system types and names
  BL buffer hash table, CF control file transaction, CI cross-instance call invocation, CU cursor bind, DF data file, DL direct loader parallel index creation, DM database mount, DR distributed recovery, DX distributed transaction, FS file set, HW space management operation, IN instance number, IR instance recovery, IS instance state, IV library cache invalidation, JQ job queue, KK redo thread kick, L[A-P] library cache lock, MM mount definition, MR media recovery, N[A-Z] library cache pin, PF password file, PI/PS parallel operation, PR process startup, Q[A-Z] row cache, RT redo thread, SC system commit number, SM smon, SN sequence number, SQ sequence number enqueue, SS sort segment, ST space transaction, SV sequence number value, TA generic enqueue, TS temporary segment (ID2=0) or new block allocation (ID2=1), TT temporary table, UN user name, US undo segment ddl, WL being-written redo log, XA instance registration attribute lock, XI instance registration lock

Table Locks (TM)

SQL Statement	Mode Acquired	Additional Mode Allowed?					Row Locks?
		RS	RX	S	SRX	X	
select	none	Y	Y	Y	Y	Y	
select … for update	RS	Y*	Y*	Y*	Y*	N	X
lock table … in row share mode	RS	Y	Y	Y	Y	N	
insert	RX	Y	Y	N	N	N	X
update	RX	Y*	Y*	N	N	N	X
delete	RX	Y*	Y*	N	N	N	X
lock table … in row exclusive mode	RX	Y	Y	N	N	N	
lock table … in share mode	S	Y	N	Y	N	N	
lock table … in share row exclusive mode	SRX	Y	N	N	N	N	
lock table … in exclusive mode	X	N	N	N	N	N	

RS = SS (subshare), RX = SX (subexclusive), SRX = SSX (share-subexclusive)
* waits occur for conflicting row locks of concurrent transactions

Database

Views & Tables

v$database, v$controlfile, v$controlfile_record_section, v$deleted_object, v$compatibility, v$compatseg, v$timezone_names, dictionary, dict_columns, dba_catalog, dba_objects, dba_object_size, dba_keepsizes, dba_analyze_objects, props$, database_properties, database_compatible_level

Parameters

db_create_file_dest, db_create_online_log_dest_<n>, undo_tablespace, cluster_database, control_files, db_name, db_domain, db_files, compatible, read_only_open_delayed

Files

catalog.sql, catproc.sql, utlrp.sql, utlip.sql, utlirp.sql, utlconst.sql, utlincmpt.sql, utldst.sql, timezone.dat, timezlrg.dat, catlg803.sql, u0703040.sql, r0703040.sql, u080<n>0<n>0.sql, r08000<n>0.sql, d080<n>0<n>.sql

Tuning/Contention

phyrds, phywrts (v$filestat)

DB Creation

create database [<db>]
    [datafile '<file>' [, …] size <n> [reuse]
    [autoextend {on | off} [next <1xBS>
        maxsize {<n> | unlimited}]]]
    [logfile [group <n>]
        ('<log>' [, …]) size <n> [reuse]
    [, [group <n>]
        ('<log>' [, …]) size <n> [reuse]] …]
    [default temporary tablespace <ts>
        [tempfile '<file>']
        [extent management local]
        [uniform [size <1> [k | m]]]]
    [undo tablespace <SYS_UNDOTBS>
        [datafile '<file>'
            [autoextend… [, …]]]
    [controlfile reuse]
    [maxdatafiles <n>] [maxinstances <n>]
    [maxlogfiles <n>] [maxlogmembers <n>]
    [maxloghistory <n>]
    [character set {<charset>
        | <UTF8> | <UTFE> | <AL32UTF8>}]

[national character set
    {<UTF8> | <AL16UTF16>}]
[set time_zone =
    { '<{+|-}hh:mi>'
    | '<time_zone_region>' }]
[set standby database
    {protected | unprotected}]
[archivelog | noarchivelog] [exclusive];

DB Modification

alter database [<db>] rename global_name
    to <db>;
alter database [<db>] default temporary
    tablespace <ts>;
alter system set undo_tablespace = <new_ts>;
alter database [<db>] convert;
alter database [<db>] reset compatibility;
alter database [<db>] [national] character set
    <new_char>;
alter database [<db>] set {dblow = <str> |
    dbhigh = <str> | dbmac {on | off} };

create controlfile ['<ctrl>'] [reuse]
    set database <db> [datafile…] [logfile…]
    … [[no]resetlogs];
create spfile [= '<spfile>'] from
    pfile [= '<pfile>'];
create pfile [= '<pfile>'] from
    spfile [= '<spfile>'];
alter database [<db>] backup controlfile to
    { '<file>' [reuse]
    | trace [resetlogs | noresetlogs] };
alter database [<db>] create standby controlfile
    as '<file>' [reuse];
alter database [<db>] set standby database
    {protected | unprotected};
alter database [<db>]
    commit to switchover to [physical]
    {primary | standby} [wait | nowait];
alter database [<db>] activate [physical]
    standby database [skip [standby logfile]];

Data Guard CLI

dgmgrl [-silent] [-xml] [-debug] [-echo]
connect <user>/<pwd>@<service>
startup [restrict] [force] [pfile=<file>]
    [nomount | mount [<db>]
    | open [read {only | write}]]
shutdown {normal | immediate | abort}
show { configuration [verbose] ['<prop>']
    | site [verbose] '<site>' ['<prop>']
    | resource [verbose] '<res>' ['<prop>']
        [on site '<site>']
    | dependency tree
    | log [alert] [latest] on site '<site>' };
enable { configuration | site '<site>' | resource
    '<res>' [on site '<site>'] };
disable { configuration | site '<site>' | resource
    '<res>' [on site '<site>'] };

alter
    { configuration set state =
        '[online | offline]'
    | site '<site>' set { state = '[online | offline]'
    | auto pfile = '<pfile>' [off] }
    | resource '<res>' [on site '<site>'] set
        { state = '<state>'
        | property '<prop>' = '<val>' }; }
create
    { configuration '<conf>'
        as primary site is '<prim>'
    | site '<site>' }
    resource is '<res>' hostname is '<host>'
    instance name is '<inst>'
    service name is '<serv>'
    site is maintained as physical;
remove { configuration '<conf>' | site '<site>' };

Other Utilities

dbassist
dbca

Database Utilities

Views & Tables

v$loadcstat, v$loadistat, v$loadpstat,
v$loadcstat, dba_exp_files, dba_exp_objects,
dba_exp_version, sys.incexp, sys.incfil,
sys.incvid

Files

catexp.sql, catexp7.sql, migrate.bsq

Export

exp

    help = <u><n></u> userid = <user>/<pwd> parfile
= <par> file = <u><expdat.dmp></u> filesize
= <n> volsize = <n> log = <log> buf-
fer = <n> silent = <u><n></u> recordlength =
<n> direct = <u><n></u> rows = <u><y></u> indexes
= <u><y></u> grants = <u><y></u> constraints = <u><y></u>
triggers = <u><y></u> feedback = <u><0></u> statistics
= {<u>estimate</u> | compute | none} record =
<u><y></u> compress = <u><y></u> consistent = <u><n></u>
object_consistent = <n> flashback_scn =
<scn> flashback_time = <time> resumable
= <u><n></u> resumable_name = <str> resum-
able_timeout = <u><7200></u> template = <x>
tablespaces = (<ts> [, …]) transport-ta-
blespace = <n> tts_full_check = <x> [, …]
point_in_time_recover = <n>
 recovery_tablespaces = <ts> [, …]
{ full = <u><n></u> | <u>owner</u> = <schema>
| tables = (<tab>[:<part>] [, …]
    [query = <expr>]) }
inctype = {complete | cumulative | incre-
mental} << deprecated
Perform full exports as user System.
buffer size =
    (n rows) * SUM(max. field length + size of length
    indicator [2 or 3, system dependent])

Import

imp

    help = <u><n></u> userid = <user>/<pwd> parfile
= <par> file = <u><expdat.dmp></u> filesize =
<n> volsize = <n> log = <log> buffer
= <n> recordlength = <n> rows = <u><y></u>
grants = <u><y></u> indexes = <u><y></u> indexfile =
<file> constraints = <u><y></u> commit = <u><n></u>
compile = <u><y></u> ignore = <n> inctype =
{system | restore} feedback = <u><0></u> show
= <n> statistics = {<u>always</u> | none | safe |
recalculate} analyze = <y> recalculate_sta-
tistics = <n> destroy = <n> skip_unus-
able_indexes = <n> toid_novalidate =
(<type> [, …]) resumable = <n> resum-
able_name = <str> resumable_timeout
= <u><7200></u> streams_configuration = <y>
streams_instatiation = <u><n></u> { full = <u><n></u> |
tables = (<tab>[:<part>] [, …])} fromuser
= <schema> [, …] touser = <schema> [,
…] transport_tablespace = <u><n></u> datafiles =
‘(<file> [, …])‘ tablespaces = (<ts> [, …])
tts_owners = <owner> [, …] point_in_
time_recover = <false>
Order: type defs – table defs – table data – indexes
          – constraints, views, procedures, triggers
          – bitmap, functional, domain indexes

Loads

sqlldr

    userid = <user>/<pwd> data = <data>
control = <cntrl> parfile = <par> log =
<log> bad = <bad> discard = <discard>
discardmax = <n> skip = <n> load =
<n> errors = <n> rows = <n> bindsize
= <u><65536></u> readsize = <u><65536></u> silent
= ({header | feedback | errors | discards
| partitions | all} [, …]) direct = <u><n></u>
multithreading = <n> streamsize = <n>
columnarrayrows = <n> parallel = <n>
file = <file> skip_unusable_indexes = <u><n></u>
skip_index_maintenance = <n> com-
mit_discontinued = <u><n></u> external_table
= {<u>not_used</u> | generate_only | execute}
resumable = <u><n></u> resumable_name = <str>
resumable_timeout = <u><7200></u> datecache
= <n>
bind array size =
    (n rows) * (SUM (fixed field lengths) + SUM(max.
    varying field lengths) + ((number of varying length
    fields) * (size of length indicator [2 or 3, system
    dependent])))

Controlfile

[options (

    [bindsize = <n>] [columnsarrayrows =
<n>] [direct = {true | false}] [errors = <n>]
[load = <n>] [multithreading = {true |
false}] [parallel = {true | false}] [readsize
= <n>] [resumable] [resumable_name]
[resumable_timeout] [rows = <n>] [silent
= ({feedback | errors | discards | all} [, …])]
[skip = <n>] [skip_index_maintenance]
[skip_unusable_indexes] [streamsize =
<n>])]

[<u>recoverable</u> | unrecoverable]

{load | continue_load} data

    [{infile | inddn} ‘<load.dat>’ | *]
        [“str [x]‘<char>’”]
    [“recsize <n> buffers <n>”]
    [badfile ‘<load.bad>’ | baddn]
    [{discardfile | discarddn} ‘<load.dsc>’]
    [{discards | discardmax} <n>]
    [characterset <char>]
    [byteorder {big | little} [endian]]
    [byteordermark {check | nocheck}]
    [length {semantics}
        {byte | char | character}]
    [concatenate <n>]
    [continueif
        { [this | next] [preserve]
            [(<pos>
       | last [preserve] [(] }
       <op> [x]‘<str>’ [)]]
    into table <tab>
       [({partition | subpartition} <part>)]
    [skip <n>]
    {insert | replace | truncate | append}
    [options (file = <db_file>)]
    [when ({<col> | <pos>})
       <op> { [x]‘<str>’ | blanks } [and…]]
    [fields
       { enclosed [by]

              [[x]‘<str>’] [and [x]‘<str>’]
    | terminated [by]
        {whitespace | [x]‘<str>’ | eof}
        [[optionally] enclosed…] }
    [trailing [nullcols]]
    [sorted indexes]
    [singlerow]
    ({ <col> { <sqlfunc> | sysdate | recnum
       | sequence [({ <n>[.<x>]
                  | max | count })] }
    | <col> [filler]
       [position ({ <x> { [: | -] <y>}
                | <u>*</u> [+<z>] }))]
       { char [(<n>)]
       | varchar [(<n>)]
       | varcharc
       | date [“<fmt>”]
       | time
       | timestamp
       | time with time zone
       | timestamp with time zone
       | interval year to month
       | interval day to second
       | integer [external] [(<n>)]
       | smallint
       | float [external]
       | double
       | byteint
       | zoned [external] (p [,<s>])
       | decimal [external] (p [,<s>])
       | raw [(<n>)]
       | varraw
       | long varraw
       | varrawc
       | graphic [(<n>)]
       | graphic external [(<n>)]
       | vargraphic [(<n>)] }
    [terminated by
       {“<str>” | whitespace}]
    [{nullif | defaultif} ({<col> | <pos>})
       <op> { [x]‘<str>’ | blanks } [and…]]
    [enclosed by ‘<chr>’ and ‘<chr>’]
    [“<sql_stmt>(:<col>)”]
    [, <col> …])
    [into table <tab> …]
[begindata…]

Migration

mig

    dbname = <db> new_dbname = <new>
pfile = <initfile> spool = <logfile> check_
only = <u><false></u> no_space_check = <u><false></u>
multiplier = <u><15></u> nls_nchar = <u><char ></u>

Tablespaces, Datafiles & Segments

Views & Tables

v$tablespace, v$datafile, v$datafile_copy, v$datafile_header, v$dbfile, v$offline_range, v$tempfile, v$temp_extent_map, v$temp_extent_pool, v$temp_space_header, v$temp_ping, v$backup, v$recover_file, v$recovery_file_status, v$recovery_log, v$recovery_progress, v$recovery_status, v$recovery_transactions, v$instance_recovery, v$fast_start_servers, v$fast_start_transactions, v$managed_standby, dba_tablespaces, dba_ts_quotas, dba_data_files, dba_temp_files, dba_segments, dba_extents, dba_free_space, dba_free_space_coalesced, dba_free_space_coalesced_tmp[1-3], ts_pitr_objects_to_be_dropped, ts_pitr_check, transport_set_violations, dba_dmt_free_space, dba_dmt_used_extents, dba_lmt_free_space, dba_lmt_used_extents, pluggable_set_check, uni_pluggable_set_check, straddling_ts_objects, ext_to_obj_view, ts$, file$, filext$, uet$, fet$, seg$

Parameters

db_block_checking, db_block_checksum, recovery_parallelism, fast_start_parallel_rollback, db_file_name_convert, log_checkpoint_timeout, log_checkpoints_to_alert, db_writer_processes, db_file_simultaneous_waits, standby_file_management, read_only_open_delayed

Packages

DBMS_REPAIR
    check_object, {skip | fix}_corrupt_blocks, dump_orphan_keys, rebuild_freelists, admin_tables, segment_fix_status
DBMS_SPACE
    unused_space, free_blocks, space_usage
DBMS_SPACE_ADMIN
    tablespace_verify, tablespace_{rebuild | relocate | fix}_bitmaps, tablespace_rebuild_quotas, tablespace_fix_segment_states, tablespace_migrate_{from | to}_local, segment_{verify | corrupt | dump | moveblocks}, segment_drop_corrupt, segment_number_{blocks | extents},
DBMS_TTS
    transport_set_check, downgrade

Deprecated Features

fast_start_io_target, log_checkpoint_interval

Desupported Features

db_block_max_dirty_target, db_file_simultaneous_writes, db_block_checkpoint_batch, parallel_transaction_recovery

TS Creation

create tablespace <ts>
    [datafile '<file>'] << only optional if
        DB_CREATE_FILE_DEST is set
    [size <n>] [reuse]
    [autoextend {off | on {next <n>}
        [maxsize {<n> | unlimited}] }]
    [,'<file>'... [autoextend...]]
    [minimum extent <n>]
    [blocksize <n> [k]]
    [default storage ([initial <5xBS>]
        [next <5xBS>] [pctincrease <50>]
        [minextents <1>]
        [maxextents <n> | unlimited}]
    [freelists <1>] [freelist groups <1>]
    [buffer_pool {default | keep | recycle}])]
    [logging | nologging]
    [permanent | temporary] [online | offline]
    [extent management
        { dictionary | local
        [autoallocate | uniform [size <1m>]] }]
    [segment space management
        {manual | auto}];
create undo tablespace <ts>
    [datafile '<file>'... [autoextend...]] << s.a.
    [extent management local]
    [uniform [size <1m>]];
create temporary tablespace <ts>
    [tempfile '<file>'... [autoextend...]] << s.a.
    [extent management local]
    [uniform [size <1m>]];
drop tablespace <ts>
    [including contents [and datafiles]
        [cascade constraints]];

TS Modification

alter tablespace <ts> add {datafile | tempfile}
    '<file>' size <n> [reuse] [autoextend...];
alter tablespace <ts> rename datafile
    '<file>' [, ...] to '<new>' [, ...];
alter tablespace <ts> { online | offline
    [normal | temporary | immediate
    | for recover] }; << deprecated
alter tablespace <ts> { read {write | only}
    | permanent | temporary };
alter tablespace <ts> [minimum extent <n>]
    default storage (...);
alter tablespace <ts> coalesce;
alter tablespace <ts> {begin | end} backup;
alter database [<db>] datafile <n> [, ...]
    end backup;

Datafiles

alter system checkpoint [global | local];
alter system check datafiles [global | local];
alter database [<db>] datafile '<file>' [, ...]
    { resize <n> | autoextend... | online
    | offline [drop] | end backup };
alter database [<db>] tempfile '<file>' [, ...]
    { resize <n> | autoextend... | online
    | offline | drop [including datafiles] };
alter database [<db>] rename file
    '<file>' [, ...] to '<new_file>' [, ...];
alter database [<db>] create datafile
    '<file>' [, ...] [as {new | '<file>' [, ...]}]];
alter system dump datafile '<file>'
    block min <x> block max <y>;

Recovery

set autorecovery {on | off}
set logsource <dir>
alter database [<db>] recover
    [automatic] [from '<log_path>']
    { { [standby] database
        [until { cancel | change <scn>
        | time '<YYYY-MM-DD:HH24:MI:SS>' }]
        [using backup controlfile]
    | managed standby database
        [{ next <n> | timeout <n> | delay <n>
        | nodelay | expire <n> } | cancel
        [immediate] [nowait] | disconnect
        [from session] [finish [nowait]]]
    | [standby] tablespace '<ts>' [, ...]
        [until [consistent with] controlfile]
    | [standby] datafile {'<file>' | <n>} [, ...]
        [until [consistent with] controlfile]
    | logfile '<log>' }
        [test [allow <x> corruption]]
        [noparallel | parallel [<n>]]
    | continue [default] | cancel };
recover [automatic] [from '<log_path>']
    { database [until { cancel | change <scn>
        | time '<YYYY-MM-DD:HH24:MI:SS>' }]
        [using backup controlfile]
    | [managed] standby database
        [timeout <n> | cancel [immediate]]
    | [standby] tablespace '<ts>' [, ...]
        [until [consistent with] controlfile]
    | [standby] datafile {'<file>' | <n>} [, ...]
        [until [consistent with] controlfile]
    | logfile <log>
    | continue [default]
    | cancel }
    [noparallel | parallel (degree {<n> | default}
        [instances <1> | default])]

Tablespaces, Datafiles & Segments (cont.)

Utilities
dbv file=<file>
    start=<n> end=<n> logfile=<log>
    blocksize=<2048> feedback=<0>
    parfile=<fil> segment_id=<ts.fil.blck>

Blocks
v$type_size
Block header:
    static (61B), table directory, row directory (2B*rec), interested transaction list (23B*tx)

Row piece:
    overhead, no. of columns, cluster key ID, rowids of chained row pieces, col data (col length, col value, …)

RowID
Logical:
    hex string of variable length
Extend(10B):
    DataObj#{32b} - RelFile#{10b} - Block#{22b} - Row#{16b}

Base64
    OOOOOO – FFF – BBBBBB – RRR
Restrict(6B):
    Block#{Xb} - Row#{Xb} - File#{Xb}

Packages
DBMS_ROWID
    (rowid_create, rowid_object, rowid_relative_fno, rowid_block_number, rowid_row_number, rowid_to_absolute_fno, rowid_to_extended, rowid_to_restricted)

Logfiles

Views & Tables
vlog, vlogfile, v$thread, v$loghist, v$log_history, v$database, v$archive, v$archive_dest, v$archive_dest_status, v$archive_gap, v$standby_log, v$archived_log, v$archive_processes, v$logmnr_dictionary, v$logmnr_parameters, v$logmnr_logs, v$logmnr_contents, dba_source_tables, dba_source_tab_columns, dba_subscriptions, dba_subscribed_tables, dba_subscribed_columns, change_sources, change_sets, change_tables

Parameters
db_create_online_log_dest_<1-5>, thread, log_buffer, log_archive_max_processes, log_archive_start, log_archive_dest, log_archive_format, standby_archive_dest, log_archive_duplex_dest, log_archive_dest_<1-10>, log_archive_dest_state_<1-10>, remote_archive_enable, fal_client, fal_server, log_archive_trace, archive_lag_target, log_archive_min_succeed_dest, log_file_name_convert, arch_io_slaves, utl_file_dir, logmnr_max_persistent_sessions, _log_simultaneous_copies, _log_io_size

(_allow_resetlogs_corruption
    << undocumented & unsupported)

Packages
DBMS_LOGMNR_D
    build
DBMS_LOGMNR
    add_logfile, start_logmnr, end_logmnr, mine_value, column_present
DBMS_LOGMNR_SESSION
    {add | remove}_logfile, {create | attach | detach | destroy}_session, column_present, include_src_tbl, mine_value, {prepare | release}_scn_range, set_dict_attr, set_session_params
DBMS_[LOGMNR_]CDC_PUBLISH
    {create | alter | drop}_change_table, drop_{subscription | subscriber_view}, purge
DBMS_[LOGMNR_]CDC_SUBSCRIBE
    get_subscription_handle, subscribe, {activate | drop}_subscription, {extend | purge}_window, {prepare | drop}_subscriber_view

Files
dbmslm.sql, dbmslmd.sql, dbmslms.sql, dbmscdcp.sql, dbmscdcs.sql catcdc.sql, initcdc.sql

Tuning/Contention
v$system_event, v$sysstat
Redo latch:
    «redo allocation», «redo copy» (v$latch)
    «misses» / «gets» > 1% or
    «immediate_misses» / («immediate_gets» + «immediate_misses») > 1%
    -> decrease «log_small_entry_max_size»
    -> increase «log_simultaneous_copies» (max. CPU * 2)

Desupported Features
v$targetrba, log_archive_buffers, log_archive_buffer_size, log_block_checksum, log_simultaneous_copies, log_small_entry_max_size, lgwr_io_slaves

Archive Mode
archive log { list | stop | {start | next | all | <n>}}
    [to <dest>] << always applies to current instance
alter database [<db>]
    {archivelog | noarchivelog};
alter system archive log [thread <n>]
    { start [to '<log_path>'] | stop
    | current << global log switch
    | next | all | sequence <n> | group <n>
    | change <n> | logfile '<file>' };
alter system switch logfile;
    << applies only to current instance

Logfiles
alter database [<db>] add [standby] logfile
    [thread <n>] [group <n>] ('<log>', …)
    size <n>;
alter database [<db>]
    {enable [public] | disable} thread <n>;
alter database [<db>] add [standby] logfile
    member '<log>' [reuse] to group <n>;
alter database [<db>] register [or replace]
    [physical] logfile '<log>' [, …];
alter database [<db>] rename file
    '<log>' [, …] to '<new_log>' [, …];
alter database [<db>] drop
    [standby] logfile group <n>;

alter database [<db>] drop
    [standby] logfile member '<log>';
alter database [<db>] clear
    [unarchived] logfile {group <n> | '<log>'}
    [unrecoverable datafile];
alter database add supplemental log data
    ({primary key | unique index} [, …])
    columns;
alter database drop supplemental log data;
alter table add supplemental log group <grp>
    (<col> [, …]) [always];
alter table drop supplemental log group <grp>;

Tables, Constraints & Triggers

Views & Tables

dba_tables, dba_all_tables, dba_object_tables, dba_external_tables, dba_external_locations, dba_tab_comments, dba_tab_columns, col, dba_tab_col_statistics, dba_associations, dba_ustats, dba_col_comments, dba_updatable_columns, dba_unused_col_tabs, dba_tab_modifications, dba_nested_tables, dba_part_tables, dba_tab_partitions, dba_tab_subpartitions, dba_part_col_statistics, dba_part_key_columns, dba_partial_drop_tables, dba_subpart_col_statistics, dba_subpart_key_columns, dba_constraints, dba_cons_columns, dba_triggers, dba_trigger_cols, dba_internal_triggers, dba_tab_histograms, dba_part_histograms, dba_subpart_histograms, tab$, col$, con$, cdef$, ccol$, trigger$, triggercol$, com$, hist_head$, tabpart$, tabsubpart$

Parameters

_system_trig_enabled

Packages

DBMS_UTILITY

analyze_database, analyze_schema, analyze_part_object

DBMS_STATS

{set | get}_{column | index | system | table}_stats, gather_{system | database | schema | table | index}_stats, delete_{system | database | schema | table | index | column}_stats, {export | import}_ {system | database | schema | table | index | column}_stats, prepare_column_values, convert_raw_value, generate_stats, {create | drop}_stat_table, alter_{database | schema}_table_monitoring, flush_{database | schema}_monitoring_info

DBMS_REDEFINITION

{can | start | finish | abort}_redef_table, sync_interim_table

Files

utlexcpt.sql, utlexcpt1.sql, dbmsstdx.sql, dbmsutil.sql, utlvalid.sql, utlchain.sql, utlchn1.sql

Tuning/Contention

pctfree = UPD/AVG

pctused = 1 − pctfree − AVG/nBLK

scans:
    «table scans%» (v$sysstat) -> adjust
    «db_file_multiblock_read_count»

row migr.:
    «table_name», «head_rowid» (chained_
    rows << utlchain.sql, utlchain1.sql) or «table
    fetch continued row» (v$sysstat)
    -> increase pctfree
    -> recreate table

freelists:
    «segment header» (v$waitstat), «buffer
    busy waits» (v$system_event)
    -> alter pctfree/pctused, inittrans or
    -> increase freelist/freelist groups
    (v$session_wait -> dba_extents -> dba_seg-
    ments -> recreate object)

full & partial partition-wise joins

Desupported Features

dba_histograms, cache_size_threshold

Table Creation

(max. 1000 col)

```
create [global temporary] table <tab>
    [of <object_type>
        [[not] substitutable at all levels]
        [object identifier is
            {primary key | system generated} ]
        [oidindex <ind> ([tablespace <ts>…]
            [storage (…)]) ] ]
    ( <col> <type> [ {default | := } <value>]
        [with rowid] [scope is <tab>]
        [constraint <col_constr>]
        [ {[not] null
        | primary key [using index
            { <ind>…
            | (create unique index <ind>
                on <tab> (<col> [,…]) …) } ]
        | unique [using index
            { <ind>…
            | (create unique index <ind>
                on <tab> (<col> [,…]) …) } ]
        | check (<expr>)
        | references <tab> [(<col>)]
            [on delete {cascade | set null}] }
        [ [not] deferrable [initially
            {immediate | deferred}] ]
        [ {enable | disable}
            [validate | novalidate]
            [exceptions into <tab>] ]
        [,<col>… [constraint <col_constr>]…]
        [, …] [, constraint <tab_constr>…]
        [ref (<col>) with rowid]
        [scope for (<col>) is <tab>]
        [supplemental log group <grp>
            (<col> [, …]) [always] ] )
    [on commit {delete | preserve} rows]
```

```
[pctfree <10>] [pctused <40>]
[initrans <1>] [maxtrans <n>]
[storage (…) ] [tablespace <ts>]
[logging | nologging]
[recoverable | unrecoverable] << deprecated
[organization
    { heap
    | index << see index-organized table below
    | external ( [type <oracle_loader>]
        default directory <dir>
        [access parameters
            { (<opaque_format>)
            | using clob <subquery> } ]
        location ([<dir>:] '<loc>' [, …]) )
        [reject limit {<0> | unlimited}] ] }
[cluster <clust> (<col> [, …])]
[column <col>
    { <element> is of [type] (only <type>)
    | [not] substitutable at all levels } ]
[nested table <col>
    [ <element> is of [type] (only <type>)
    | [not] substitutable at all levels ]
    store as <tab>
    [( (<prop>) [storage (…)] )]
    [return as {locator | value}] ]
[varray <varr>
    [ <element> is of [type] (only <type>)
    | [not] substitutable at all levels ]
    store as lob <lobseg>
        [([tablespace <ts>]…)] ]
[lob { (<col>) store as <lob_seg>
    | (<col> [, …]) store as }
    ( [tablespace <ts>] [storage (…)]
    [{enable | disable} storage in row]
    [pctversion <10>] [chunk <n>]
    [cache | {nocache | cache reads}
```

```
        [logging | nologging] ]
        [index <ind> << deprecated in 8i and 9i
            ([tablespace <ts>]
            [storage (…)]) ]
        [tablespace <ts>… ] ) ]
[XMLType [<col>] store as clob [<lobseg>]
    [([tablespace <ts>]…)] ]
[ partition by range (<col> [, …])
    [subpartition by hash (<col> [, …])
        [subpartitions <n>
        [store in (<ts> [, …])] ]
    ( partition [<part>]
        values less than
        ( [<val> [, …] | maxvalue} )
        [storage (…)] [tablespace <ts>]
        [ (subpartition <subpart>
                [tablespace <ts>]
        [, subpartition…] )
        | subpartitions <n>
            store in (<ts> [, …]) ]
    [, partition…
        [(subpartition…)] ] )
| partition by list (<col>)
    (partition [<part>]
        values ({<val> [, …] | null})
        [storage (…)] [tablespace <ts>]
        [ (subpartition <subpart>
                [tablespace <ts>]
        [, subpartition…] )
        | subpartitions <n>
            store in (<ts> [, …]) ]
    [, partition…
        [(subpartition…)] ] )
| partition by hash (<col> [, …])
    { (partition <part>
        [tablespace <ts>]
```

Tables, Constraints & Triggers (cont.)

```
        [, partition…])
    | partitions <n>
        store in (<ts> [, …]) } ]
[{disable | enable} row movement]
[cache | nocache]
[rowdependencies | norowdependencies]
[monitoring | nomonitoring]
[parallel [<n>] | noparallel]
[{enable | disable} {validate | novalidate}
  { primary key | unique (<col> [, …])
  | constraint <constr> } [using index…]
  [exceptions into <tab>] [cascade]
  [{keep | drop} index] ]
[as <subquery>];
```

Table Modification

```
alter table <tab> modify (<col> <type>…);
alter table <tab> add (<col> <type>…);
alter table <tab> set unused
    {(<col> [, …]) | column <col>}
    [cascade constraints] [invalidate];
alter table <tab> drop
    {(<col> [, …]) | column <col>}
    [cascade constraints] [invalidate]
    [checkpoint <512>];
alter table <tab> drop
    {unused columns | columns continue}
    [checkpoint <512>];
drop table <tab> [cascade constraints];
rename <tab> to <new_tab>;
alter table <tab> move
    [tablespace <ts>] [storage (…)]
    [logging | nologging]
    [parallel [<n>] | noparallel];
truncate table <tab>
    [[preserve | purge] snapshot log]
    [{drop | reuse} storage];
alter table <tab> [storage (…)]
    [noparallel | parallel [<n>] ] …
    [{nominimize | minimize}
        records_per_block];
alter table <tab>
    { allocate extent ( [ [size <n>]
        [datafile '<file>'] [instance <n>] )
    | deallocate unused [keep <n>] };
lock table <tab> in
    { row share | share update
    | row exclusive
    | share
    | share row exclusive
    | exclusive } mode [nowait];
alter table <tab> {enable | disable} table lock;
comment on {table <tab> | column
    <tab>.<col>} is '<str>';
alter table add supplemental log group <grp>
    (<col> [, …] ) [always];
alter table drop supplemental log group <grp>;
```

Partitioning

```
alter table <tab> add partition <range_part>
    values less than (<value> [, …] )
    [tablespace <ts>]
    [{update | invalidate} global indexes]
    [parallel [<n>] | noparallel];
```

```
alter table <tab> add partition
    [<hash_part> [tablespace <ts>]] […];
alter table <tab> drop partition <part>
    [, …] […];
alter table <tab> coalesce partition […];
alter table <tab> truncate
    {partition | subpartition} <part>
    [{drop | reuse} storage] […];
alter table <tab> rename
    {partition | subpartition} <part> to <new>;
alter table <tab> modify
    {partition | subpartition} <part>
    [storage (…)] [allocate extent…]
    [logging | nologging …
    [[rebuild] unusable local indexes];
alter table <tab> modify
    partition <part>
    {add subpartition
        [<subpart> [tablespace <ts>]]
    | coalesce [subpartition] } […];
alter table <tab> modify default attributes
    [for partition <comp_part>]
    [storage (…)] …;
alter table <tab> modify
    partition <part>
    {add | drop} values (<val> [, …]);
alter table <tab> move
    {partition | subpartition} <part>
    tablespace <ts>
    [logging | nologging] […];
alter table <tab> split
    partition <part1> at (<n>)
    into (partition <part2>,
        partition <part3> [, …] ) […];
alter table <tab> merge partitions <part1>,
    <part2> [into partition <part3>] […];
alter table <tab> exchange
    {partition | subpartition} <part>
    with table <tab> [including indexes]
    [{with | without} validation] […];
```

Constraints

```
alter table <tab> add
    ( [constraint <tab_constr>]
    { primary key (<col> [, …])
        [using index…]
    | unique (<col> [, …]) [using index…]
    | foreign key (<col> [, …])
        references <tab>  [(<col> [, …])]
        [on delete {cascade | set null} ]
    | check (<expr>) }
    [[not] deferrable
        [initially {immediate | deferred}] ]
    [{disable | enable} {validate | novalidate}
        [exceptions into <tab>]] );
alter table <tab>
    {disable | enable} {validate | novalidate}
    { constraint <constr>
    | primary key
    | unique (<col> [, …]) }
    [using index…]
    [exceptions into <tab>] [cascade]
    [{keep | drop} index];
alter table <tab> modify constraint <constr>
```

```
    … [rely | norely];
alter table <tab> drop
    { constraint <constr> [cascade]
    | { primary key | unique (<col> [, …]) }
        [cascade] [{keep | drop} index] };
set constraint[s] {<constr> [, …] | all}
    {immediate | deferred};
```

Triggers

```
alter table <tab> {enable | disable} all triggers;
create [or replace] trigger <trigg>
    { before | after | instead of }
    { {delete | insert | update [of <col> [, …]] }
        [or…] on {<tab> | [nested table <col>
                        of] <view>}
    | { { {associate | disassociate} statistics
        | analyze | audit | noaudit | comment
        | create | alter | rename | drop | grant
        | revoke | truncate | ddl } [or…]
    | { shutdown | startup | servererror
        | logon | logoff } [or…] }
      on {schema | database} }
    [referencing {old [as] <old> | new [as]
        <new> | parent [as] <parent>} [, …] ]
    [for each row] [when (<expr>)]
    { begin        <stat>;        end;
    | call … ; }
alter trigger <trigg>
    {enable | disable | compile [debug]};
drop trigger <trigg>;
```

Statistics

```
deprecated (use DBMS_STATS) >>
analyze table <tab>
    [partition (<n>) | subpartition (<n>)]
    { compute [system] statistics
    | estimate [system] statistics
        [sample <1064> {rows | percent}] }
    [for table] [for all [local] indexes]
    [for all [indexed] columns [size <75>] ]
    [for columns [size <75>]
        <col> [size <75>] [<col>…] ];
analyze table <tab> delete [system] statistics;
analyze table <tab> list chained rows
    [into <chained_rows>];
analyze table <tab> validate
    { structure [cascade] [into <invalid_rows>]
        {online | offline}
    | ref update [set dangling to null] };
associate statistics with
    { columns <tab>.<col> [, …]
    | functions <func> [, …]
    | packages <pack> [, …]
    | types <type> [, …]
    | indexes <ind> [, …]
    | indextypes <indtype> [, …] }
    [using <stat_func>]
    [default cost (<cpu>, <io>, <network>)]
    [default selectivity <selec>];
disassociate statistics from
    { columns <tab>.<col> [, …]
    | functions <func> [, …]
    | packages <pack> [, …]
    | types <type> [, …]
```

Tables, Constraints & Triggers (cont.)

```
| indexes <ind> [, …]
| indextypes <indtype> [, …] }
[force];
```

External Table Opaque Format

```
record
    { {fixed | variable} <n>
    | delimited by {newline | '<str>'} }
    [characterset '<char>']
    [data is {little | big} endian]
    [string sizes are in {bytes | characters} ]
    [load when <expr>]
    [{badfile [<dir>:] '<file>' | nobadfile]
    [discardfile [<dir>:] '<file>' | nodiscardfile]
    [logfile [<dir>:] '<file>' | nologfile]
    [skip <n>]
    [fields
        [ enclosed by '<str>' [and '<str>']
```

```
| terminated by { whitespace | '<str>'}
        [[optionally] enclosed by…] ]
[ltrim | rtrim | lrtrim | ldtrim | notrim]
[missing field values are null]
[(<field>
    [ [position] ({ * | <start> | [+|-]
        <incr>} [:|-] {<end> | <len> }) ]
    [ [unsigned] integer [external] <n>
    | {decimal | zoned} [external]
                (<p> [, <s>])
    | oracle date
    | oracle number [counted]
    | {double | float} [external]
    | raw <n>
    | char <n> [enclosed…] [ltrim…]
      [dateformat
        { { date | {time | timestamp}
            [with timezone] }
        mask "<fmt>"
```

```
| interval { year_to_month
            | day_to_second } } ]
| { varchar | varraw | varcharc
    | varrawc ([<n>,] <max>) ]
[{defaultif | nullif} <expr>]
[, <field> …] ) ]
```

Views, Synonyms & Sequences

Views & Tables

dba_views, dba_synonyms, dba_sequences

Views

create [or replace] [force | <u>noforce</u>]
    view <view>
    [({ <alias> [<col_constr>]
        | <constr> } [, …])
    | of <type>
        { with object identifier
          [default | (<attr>, …)]
        | under <superview> }
        ({ <attr> <col_constr>
        | <constr> } [, …])]
    as <query>

```
[with { read only | check option
        [constraint <constr>} } ];
alter view <view> <constr>…;
```
View constraints are declarative only. Only unique or prim./foreign key with mode disable novalidate.
```
alter view <view> compile;
rename <view> to <new_view>;
drop view <view>;
```

Synonyms

```
create [public] synonym <syn> for <obj>;
rename <syn> to <new_syn>;  << only private!
drop [public] synonym <syn>;
```

Sequences

```
create sequence <seq>
    [start with <1>] [increment by <1>]
    [maxvalue <10^27> | nomaxvalue]
    [minvalue <1> | nominvalue]
    [cycle | nocycle] [nocache | cache <20>]
    [order | noorder];
```
When an instance shuts down, cached sequence values that have not been used in committed DML statements can be lost.
Ordered sequences may not be cached with RAC.
```
alter sequence <seq> …;
rename <seq> to <new_seq>;
drop sequence <seq>;
```

Clusters

Views & Tables

dba_clusters, dba_clu_columns, all_tab_columns, dba_cluster_hash_expressions

Creation & Modification

create cluster <clust>
    (<col> <type> [, …])
    [<u>index</u> | {single table} hashkeys <n>
           [hash is <expr>]]

```
[size <1xBS>]
[tablespace <ts>] [storage (…)]
[pctfree <10>] [pctused <40>]
[initrans <n>] [maxtrans <255>];
create index <ind>
    on cluster <clust> [pctfree <n>]
    [tablespace <ts>] [storage (…)]
    [initrans <n>] [maxtrans <n>];
create table <tab>
    (<col> <type>… [constraint <constr>…] )
```

```
cluster <clust> (<col> [, …] );
alter cluster <clust>…;
truncate cluster <clust>
    [ {<u>drop</u> | reuse} storage];
drop cluster <clust>
    [including tables [cascade constraints]];
analyze cluster <clust> …;
```

Index-organized Tables

Views & Tables

all_tables (iot_type, iot_name), all_indexes

Creation & Modification

create table <iot>
    (<col>… primary key…)
    organization index
    [tablespace <ts>] [storage (…)]
    [pctfree <n>] [initrans <n>] [maxtrans <n>]
    [mapping table | nomapping]
    [pctthreshold <50> [including <col>]]
    [compress [<n>] | nocompress]

[overflow [tablespace <ts>] [pctfree <10>]
    [initrans <1>] [maxtrans <255>]
    [storage (…)]
    [allocate…] [deallocate…]
    [logging | nologging]]
[partition by range (<col> [, …])
    (partition <partX>
        values less than (<value> [, …])
        [storage (…)] [tablespace <ts>]
        [overflow tablespace <ts>…]
    [, partition…])];
alter table <iot> … [overflow…];
alter table <iot> add overflow …

    [(partition <part>…)];
alter table <iot> move [online]
    [compress [<n>] | nocompress]
    [tablespace <ts>] [overflow…] …
    [noparallel | parallel [<n>]];
alter table <iot> modify default attributes
    [for partition <part>] [storage (…)]
    [pctthreshold <50> [including <col>]]
    [compress [<n>] | nocompress]
    [overflow tablespace <ts>…];
alter table <iot> coalesce;
analyze table <iot> compute statistics;

Indexes

Views & Tables

v$object_usage, dba_indexes, dba_indextypes, dba_indextype_operators, dba_ind_columns, dba_ind_expressions, index_stats, dba_part_indexes, dba_ind_partitions, dba_ind_subpartitions, dba_part_col_statistics, dba_subpart_col_statistics, index_histogram, ind$,

icol$, icoldep$

Parameters

create_bitmap_area_size, bitmap_merge_area_size

Packages & Files

DBMS_PCLXUTIL
    build_part_index

Tuning/Contention

index_stats:
    «del_lf_rows_len» / «lf_rows_len» > 20%

Index Creation

create [unique | bitmap]
    index <ind> on <tab>
    { ([<expr>] <col> [asc | desc] [, …])
    | ([<tab>] <col> [asc | desc] [, [<tab>]…])
      from <tab> [, <tab>…] where <expr> }
    [tablespace {<ts> | default }]
    [storage (…)] [pctfree <10>]
    [initrans <n>] [maxtrans <255>]
    [logging | nologging] [nosort] [reverse]
    [online] [noparallel | parallel [<n>]]
    [compress [<n>] | nocompress]
    [local
        [(partition [<partX>] [storage (…)]

on range partitioned table
        [tablespace <ts>]
        [logging | nologging]
        [, partition…])
    | [store in ({<ts> [, …] | default})
      | (partition [<partX>]

on hash partitioned table
      [tablespace <ts>]
      [, partition…])]
    | store in ({<ts> [, …] | default})
      [(partition [<partX>]
        [storage (…)]

on composite partitioned table
      [tablespace <ts>]
      [logging | nologging]
      [store in ({<ts> [, …] | default})
      | (subpartition [<subpartX>]
        [tablespace <ts>]
        [, subpartition…])]
    [, partition…])]]]
    [global partition by range (<col>)
      (partition <partX>

        values less than
        ({<value> [, …] | maxvalue})
        [storage (…)] [tablespace <ts>]
        [logging | nologging]
    [, partition…])]
    [indextype is <type>
      [parameters ('<str>')]];
drop index <ind>;
alter index <ind> {enable | disable};
alter index <ind> unusable;
alter index <ind> rename to <new>;

Index Partitioning

alter index <ind> drop partition <part> [, …];
alter index <ind> rename
    {partition | subpartition} <part> to <new>;
alter index <ind> modify
    {partition | subpartition} <part>
    [storage (…)] …
    [logging | nologging] [unusable]
    [rebuild unusable local indexes];
alter index <ind> modify default attributes
    [for partition <part>]
    [storage (…)] [pctfree <n>] …;
alter index <ind> rebuild
    {partition | subpartition} <part>
    [tablespace <ts>] [parallel [<n>]];
alter index <ind> split partition <p1>
    at values less than (<n>) into
    (partition <p2>, partition <p3> [, …]);

Index Modification

alter index <ind> [storage (…)]
    [initrans <n>] [maxtrans <n>]
    [compress [<n>] | nocompress];
alter index <ind>
    { allocate extent ([size <n>]
        [datafile '<file>'] [instance <n>])
    | deallocate unused [keep <n>] };
alter index <ind> rebuild
    [{partition | subpartition} <part>]
    [tablespace <ts>] [storage (…)]
    [pctfree <10>]
    [initrans <n>] [maxtrans <255>]
    [logging | nologging]
    [parallel [<n>] | noparallel]
    [compress <n> | nocompress]
    [compute statistics] [online]
    [reverse | noreverse]
    [parameters ('<par>')];
alter index <ind> coalesce;

Statistics

analyze index <ind>…;
analyze index <ind> validate structure
    {online | offline};
alter index <ind>
    {monitoring | nomonitoring} usage;

Undo Management

Views & Tables

v$undostat, v$rollname, v$rollstat, v$transaction, v$transaction_enqueue, v$global_transaction, dba_undo_extents, dba_rollback_segs, dba_pending_transactions

Parameters

undo_management, undo_tablespace, undo_retention

Deprecated Features

rollback_segments, transactions, transactions_per_rollback_segment

(_corrupted_rollback_segments
 << undocumented & unsupported)

Packages

DBMS_TRANSACTION
    use_rollback_segment

Tuning/Contention

RBS Header:
    «undo segment tx slot» (v$system_event)
    > 0 or (v$rollstat) sum(«waits») /
    sum(«gets») > 5% -> add RBS
RBS Segment:
    «%undo%» (v$waitstat) / «consistent gets»
    (v$sysstat) (count/value) > 1% -> add RBS

RBS Creation

create [public] rollback segment <rbs>
    [tablespace <ts>]

[storage ([initial <5xBS>] [next <5xBS>]
    [optimal <null>] [minextents <1>]
    [maxextents {<n> | unlimited}])];
drop rollback segment <rbs>;

RBS Modification

alter rollback segment <rbs> {online | offline};
alter rollback segment <rbs> storage (…);
alter rollback segment <rbs> shrink [to <n>];
set transaction use rollback segment <rbs>;

Undo Management

create undo tablespace <ts>…;
alter system set undo_tablespace = <ts>;

Temporary Segments

Views & Tables

v$tempseg_usage, v$sort_segment, v$sort_usage, dba_segments

Parameters

sort_area_size, sort_area_retained_size

Desupported Features

sort_multiblock_read_count, sort_direct_writes, sort_write_buffers, sort_write_buffer_size

Tuning/Contention

Sorts:
    «sorts (disk)», «sorts (memory)», «sorts
    (rows)» (v$sysstat) disk.value / mem.value
    > 5% -> increase «sort_area_size»
    (+ decrease «sort_area_retained_size»)

Users, Privileges, Resources & Policies

Views & Tables

v$enabledprivs, v$resource, v$resource_limit, v$pwfile_users, v$context, v$rsrc_plan, v$rsrc_plan_cpu_mth, v$rsrc_consumer_group, v$rsrc_consumer_group_cpu_mth, v$parallel_degree_limit_mth, v$max_active_sess_target_mth, v$vpd_policy, dba_users, dba_roles, dba_profiles, dba_ustats, dba_ts_quotas, dba_sys_privs, dba_tab_privs, dba_col_privs, dba_role_privs, role_sys_privs, role_tab_privs, role_role_privs, user_tab_privs_made, user_tab_privs_recd, user_col_privs_made, user_col_privs_recd, user_password_limits, user_resource_limits, session_privs, session_roles, dba_context, dba_policies, proxy_users, resource_cost, dba_rsrc_plans, dba_rsrc_plan_directives, dba_rsrc_consumer_groups, dba_rsrc_consumer_group_privs, dba_rsrc_manager_system_privs, user$, user_history$, sysauth$, objauth$

Parameters

o7_dictionary_accessibility, remote_os_authent, os_roles, remote_os_roles, max_enabled_roles, resource_limit, resource_manager_plan, ent_domain_name

Environment

$ORA_ENCRYPT_LOGIN

Packages

DBMS_RESOURCE_MANAGER
set_initial_consumer_group, {create | submit | clear | validate}_pending_area, {create | update | delete}_{plan | plan_directive | consumer_group}, delete_plan_cascade, switch_consumer_group_for_{sess | user}
DBMS_RESOURCE_MANAGER_PRIVS
{grant | revoke}_system_privilege, {grant | revoke}_switch_consumer_group
DBMS_SESSION
switch_current_consumer_group
DBMS_RLS
{add | drop | enable | refresh}_policy, {add | drop | enable | disable | refresh}_grouped_policy, {add | drop}_policy_context, {create | delete}_policy_group

---

Users

create user <user> identified
    { by <pwd>
    | by values '<crypt_pw>'
    | externally
    | globally as '<user>' }
    [default tablespace <ts>]
    [temporary tablespace <ts>]
    [quota {<n> | unlimited} on <ts>]
    [quota…]
    [password expire]
    [account {lock | unlock}]
    [profile {<prof> | default}];
alter user <user>…;
drop user <user> [cascade];

Roles

create role <role>
    [not identified
    | identified
        { by <pwd> | using <package>
        | externally | globally }];
alter role <role>…;
drop role <role>;
alter user <user> default role
    { <role> [, …]
    | all [except <role> [, …]]

| none};
set role
    { <role> [identified by <pwd>]
        [, <role> [identified by <pwd>] …]
    | all [except <role> [, …]]
    | none };

Privileges

grant {<priv> [, …] | <role> [, …] | all
    [privileges]} to
    {<user> [, …] | <role> [, …] | public}
    [identified by <pwd>]
    [with admin option];
revoke {<priv> | <role>} from
    {<user> | <role> | public};
grant {<priv> [(<col> [, …])] [, …] | all }
    on <object>
    to { <user> [, …] | <role> [, …] | public }
    [with grant option]
    [with hierachy option];
revoke {<priv> [(<col> [, …])] | all [privileges]}
    on [directory] <object>
    from { <user> | <role> | public }
    [cascade constraints];

Profiles

create profile <prof> limit
    [{ sessions_per_user
        | cpu_per_session
        | cpu_per_call
        | connect_time
        | idle_time
        | logical_reads_per_session
        | logical_reads_per_call
        | composite_limit
        | private_sga
        | failed_login_attempts
        | password_lock_time
        | password_life_time
        | password_grace_time
        | password_reuse_time
        | password_reuse_max }
        {<n> | unlimited | default} [, …]]
    [password_verify_function
        {<func> | null | default}];
alter profile <prof> limit…;
drop profile <prof> [cascade];
alter resource cost
    [connect_time <n>] [cpu_per_session <n>]
    [logical_reads_per_session <n>]
    [private_sga <n>];

Auditing

Views & Tables

all_def_audit_opts, dba_stmt_audit_opts, stmt_audit_option_map, dba_priv_audit_opts, dba_obj_audit_opts, user_tab_audit_opts, dba_audit_trail, dba_audit_session, dba_audit_statement, change_password, dba_audit_object, dba_audit_exists, dba_audit_policies, dba_fga_audit_trail, audit_actions, sys.aud$, sys.fga_log$

Packages

DBMS_FGA
{add | drop | enable | disable}_policy

Parameters

audit_trail, transaction_auditing

Files

cataudit.sql, catnoaud.sql

SQL

[no]audit
{<stat> [, …] | <priv> [, …] }
[by <user> [, …]] [by {session | access}]
[whenever [not] successful];

[no]audit <stat> [, …] on {<object> | default}
[by {session | access}]
[whenever [not] successful];
shortcuts: user, table, procedure, resource, connect, dba, …

Net Services

Stack

Application, Server – OCI (UPI), OPI, NPI – TTC – TNS (NI,NR,NN,NS,NA) – OPA (NT) [–Protocol]

Service Name Resolution

local naming, host naming, external naming, centralized naming

Utilities

lsnrctl
{ start | stop | status | reload | set | show
| help | version | change_password
| services | save_config | trace
| dbsnmp_start | dbsnmp_stop
| dbsnmp_status }
[<LISTENER>]

agentctl
{ { start | stop | status | restart } [<agent>]
| { start | stop | status} blackout [<target>]
[-d[uration] <[d] hh:mi>]
[-s[ubsystem] <subsys>] }

namesctl
{ startup | shutdown | start | stop | reload |
restart | status | ping <ns> | reorder_ns
| start_client_cache | delegate_domain
| domain_hint | flush | flush_name
| load_tnsnames | dump_tnsnames
| dump_ldap | log_stats | reset_stats | help
| password | register | unregister | query
| timed_query | repeat | set | show
| version }

cmctl
{ start | stop | status | version }
[cman | cm | adm]

trcasst [-o[c|d]{u[q]|t} -r[0|1|2] -s -p …] <file>
netasst, tnsping, trcroute, adapters

oerr <tns> <errno>

ldapmodify

listener.ora

<LISTENER> =
(description_list = (description = (address_list = (address = (protocol = <tcp>)
(host = <node>) (port = <1521>) (key =
<prog>))) (protocol_stack = (presentation
= {ttc | giop}) (session = {ns | raw}))))

sid_list_<LISTENER> =
(sid_list = (sid_desc = (global_dbname =
<n>) << disables TAF with RAC (oracle_home
= <path>) (sid_name = <SID>) (sdu =
<n>) (program = <prog>) (prespawn_max
= <n>) (prespawn_list = (prespawn_desc
= (protocol = <n>) (pool_size = <n>)
(timeout = <n>))))),
>> Since release 8.1 sid_list_<LISTENER> only required with Enterprise Manager! <<

service_list_<LISTENER> = <n>
passwords_<LISTENER> = <n>
connect_timeout_<LISTENER> = <n>
use_plug_and_play_<LISTENER> = <n>
save_config_on_stop_<LISTENER> = <n>
trace_{level | file | directory}_<LISTENER>=<n>
logging_<LISTENER> = <n>
log_{file | directory}_<LISTENER> = <n>
startup_wait_time_<LISTENER> = <n>
queuesize = <n>
ssl_client_authentication = <n>
ssl_version = undetermined

Net Services (cont.)

tnsnames.ora (Local Naming)

```
<net_serv> =
    (description =
    (address_list =
        (failover = {on | off})
        (load_balance = {on | off})
        (source_route = {on | off})
    (address =
        (protocol = <n>) (port = <n>)
        (host = <node>)) […])
    (connect_data =
        (service_name = <serv>)
        (instance_name = <sid>)
        (handler_name = <n>) (sdu = <n>)
        (server = dedicated) (hs = ok)
        (rdb_database = <rdbfile>)
        (type_of_service = <n>)
        (global_name = <rdb>)
        (failover_mode =
                (type = {select | session | none})
                (method = {basic | preconnect})
                (retries = <5>) (delay = <1>)
                (backup = <serv>)
                (instance_role =
                    {primary | secondary | any}) ))
```
desupported:
   (connect_data = (sid = <n>))
>> Exception! Use of OEM and OPS on WinNT.
Create net service names '<SID>_startup'. <<

sqlnet.ora

log_{file | directory}_{client | server} = <n>,
use_cman = <n>, use_dedicated_server = <n>,
sqlnet.expire_time = <n>, sqlnet.{encryption
| crypto_checksum}_{client | server} =
{accepted | rejected | requested | required},
sqlnet.{encryption | crypto_checksum}_types_
{client | server} = <n>, sqlnet.crypto_seed =
<n>, trace_unique_client = <n>, trace_{level
| file | directory | timestamp}_{client | server}
= <n>, tnsping.trace_{level | directory} =
<n>, daemon.trace_{level | directory | mask}
= <n>, sqlnet.authentication_services =
<n>, sqlnet.client_registration = <n>, be-
queath_detach = <n>, disable_oob = <n>,
names.directory_path = ({hostname |
tnsnames | onames | cds | nds | nis} , …),
names.default_domain = <n>, name.default_
zone = <n>, names.preferred_servers =
<n>, names.initial_retry_timeout = <n>,
names.request_retries = <n>, names.max_
open_connections = <n>, names.message_
pool_start_size = <n>, names.dce.prefix
= <n>, names.nis.meta_map = <n>,

namesctl.internal_encrypt_password = <n>,
namesctl.internal_use = <n>, namesctl.no_ini-
tial_server = <n>, namesctl.noconfirm =
<n>, namesctl.server_password = <n>,
namesctl.trace_{level | file | directory | unique}
= <n>
desupported: automatic_ipc

names.ora   << deprecated

names.server_name = <n>, names.addresses
= <n>, names.region_checkpoint_file = <n>,
default_domain = <n>, forwarding_available =
<n>, log_file_name = <n>, log_stats_interval =
<n>, reset_stats_interval = <n>, cache_check-
point_interval = <n>, requests_enabled = <n>,
server = <n>, namesctl_trace_level = <n>,
trace_file_name = <n>, trace_level = <n>,
names.trace_{file | directory | unique} = <n>,
names.log_{file | directory} = <n>, queuesize
= <n>
desupported:
   names.use_plug_and_play, names.{domain | topol-
   ogy}_checkpoint_file

protocol.ora   << desupported

<prot>.{excluded | invited}_nodes = <node>,
<prot>.validnode_checking = <n>, tcp.nodelay
= <n>

cman.ora

```
cman =
    (address = (protocol = <tcp>)
        (host = <node>) (port = <1630>))
cman_admin = (address = (protocol = <tcp>)
        (host = <node>) (port = <1830>))
cman_profile =
    (maximum_relays = <n>, relay_statistics
    = <n>, log_level = <n>, tracing = <n>,
    trace_directory = <path>, show_tns_info
    = <n>, use_async_call = <n>, authentica-
    tion_level = <n>)
cman_rules =
    (rule_list = (rule = (src = <src>)
                (dst = <dst>) (srv = <serv>)
                (act = accept | reject)))
```

ldap.ora

Other Files

ckpcch.ora, sdns.ora, namesini.sql,
namesupg.sql, snmp_ro.ora, snmp_rw.ora,
services.ora

Environment

$TNS_ADMIN

Recovery Manager

Views & Tables

rc_database, rc_database_incarnation, rc_backup_set, rc_backup_piece, rc_checkpoint, rc_tablespace, rc_datafile, rc_backup_datafile, rc_datafile_copy, rc_proxy_datafile, rc_offline_range, rc_backup_controlfile, rc_controlfile_copy, rc_proxy_controlfile, rc_redo_log, rc_redo_thread, rc_backup_redolog, rc_archived_log , rc_log_history, rc_stored_script, rc_stored_script_line, rc_backup_corruption, rc_copy_corruption, rc_resync, v$backup_set, v$backup_piece, v$backup_datafile, v$datafile_copy, v$proxy_datafile, v$offline_range, v$backup_redolog, v$proxy_archivedlog, v$backup_device, v$backup_corruption, v$copy_corruption, v$backup_async_io, v$backup_sync_io, v$session_longops, v$session_wait

Parameters

backup_tape_io_slaves, disk_asynch_io, tape_asynch_io, control_file_record_keep_time

Packages

DBMS_BACKUP_RESTORE
DBMS_RCVCAT
DBMS_RCVMAN

Files

catrman.sql, prgrmanc.sql, dbmssbkrs.sql, prvtbkrs.plb, dbmsrman.sql, prvtrmns.plb

Desupported Features

db_file_direct_io_count, arch_io_slaves, backup_disk_io_slaves, large_pool_min_alloc

Environment

rman
    [target '<user>/<pwd>@<target_db>']
    [catalog '<user>/<pwd>@<repos_db>'
    | nocatalog]
    [auxiliary '<user>/<pwd>@<aux_db>']
    [{cmdfile [=] | @} <file>]
    [log [=] <file> [append]] [msgno]
    [trace [=] '<file>'] [debug]
    [send [=] '<cmd>']
set dbid [=] <target_dbid>;
connect {target | catalog | auxiliary}
    <user>/<pwd>@<db>
startup [nomount | mount] [force] [dba]
    [pfile [=] <file>];
shutdown [normal | transactional
    | immediate | abort];
{mount | open} database;
alter database {mount | open};
host ['<cmd>'];
debug {on | off};
set echo {on | off};
set command id to '<id>';
configure
    { snapshot controlfile name to '<file>'
    | controlfile autobackup
        { on | off | clear
        | format for device type <dev>
            {to '<fmt>' | clear} }
    | {archivelog | datafile} backup copies
        for device type <dev> {to <x> | clear}
    | default device type to <dev>
    | device type <dev> parallelism <n>
    | channel <n> device type <dev>
        connect '<user>@serv>'
    | retention policy to
        { recovery window of <x> days
        | redundancy <1> | none | clear }
    | backup optimization {on | off | clear}
    | exclude tablespace <ts> [clear]
    | maxsetsize {to {<x>| unlimited} | clear} };
show
    { retention policy
    | [default] device type
    | [auxiliary] channel
        [for device type <dev>]
    | maxset size
    | {datafile | archivlog} backup copies
    | backup optimization
    | snapshot controlfile name
    | auxname
    | exclude
    | controlfile autobackup [format]
    | all };
set snapshot controlfile name to '<file>';
send [channel <chann> [, …]]
    | device type <dev> [, …]]
    '<media_man_cmd>'
    [parms [=] '<par>'];
{create | replace} script <script> {<stat>;…}
delete script <script>;
print script <script>;
run {<cmd>; …}
run {execute script <script>;}
sql '<stat>' [' ' <file> ' '] ';

Catalog

create catalog [tablespace <ts>];
upgrade catalog [tablespace '<ts>'];
configure compatible = <n>;
drop catalog;
register database;
reset database [to incarnation <id>];
resync catalog [from controlfilecopy ['<ctrl>']];
catalog {archivelog | datafilecopy
    | controlfilecopy} '<file>' [, …]
    [tag [=] '<tag>' | level [=] <n>];
change {archivelog | datafilecopy | backuppiece | backupset | proxy | controlfilecopy}
    {'<file>' | <n> | all | tag [=] '<tag>'}
    { delete | available | unavailable | uncatalog
    | validate | crosscheck };
crosscheck

{backup | copy} [of
    { {datafile | tablespace
        | database [skip tablespace] } '<name>'
    | controlfile | archivelog
        { all | like '<name>' | {from | until}
        { time [=] '<date>' | scn [=] <n>
        | logseq [=] <n> [thread = <n>]} }]
    [tag = '<tag>'
    | completed
        { {after | before} [=] '<date>'
        | between '<date>' and '<date>' }];
delete [noprompt]
    { [expired]
        { {backup | copy} [of
            { { datafile | tablespace | database
                [skip tablespace] } '<name>'
            | controlfile
            | archivelog { all | like '<name>' |
                {from | until} { time [=] '<date>'
                | scn [=] <n> | sequence [=] <n>
                [thread = <n>]} }]
            [tag = '<tag>' | completed
                { {after | before} [=] '<date>'
                | between '<date>' and '<date>' }]
        | { {backuppiece | proxy} …
            | backupset …
            | {controlfilecopy | datafilecopy} …
            | archivelog … }
        | obsolete
            [redundancy [=] <x>
            | recovery window of <x> days
            | orphan] };
set maxcorrupt for datafile {'<file>' | <n>}
    to <n>;

Channels

allocate [auxiliary] channel <chann>
    [for {delete | maintenance}]
    { type [=] {disk | '<dev>'}
    | name [=] '<name>' }
    [parms [=] '"<par>"] [format [=] '<fm>']
    [connect [=]
        '<user>/<pwd>@<target_ops_inst>']

Recovery Manager (cont.)

```
    [debug [=] <n>] [trace [=] <n>];
set limit channel <chann> [read rate [=] <n>]
    [kbytes [=] <n>] [maxopenfiles [=] <n>];
release channel [<chann>];
```

Reporting

```
report
    { { need backup { {incremental | days}
                | redundancy } [=] <n>
      | unrecoverable }
        { datafile {'<file>' | <n>} [, …]
        | tablespace '<ts>' [, …]
        | database [skip tablespace '<ts>'
                    [, …]] }
    | obsolete  { redundancy [=] <n>
                | recovery window of <x> days
                | orphan
                | until
                        { time [=] '<date>'
                        | scn [=] <n>
                        | logseq [=] <n>
                          [thread [=] <n>] }}
    | schema [at
        { time [=] '<date>'
        | scn [=] <n>
        | logseq [=] <n> [thread [=] <n>] }] }
    [device type {disk | '<dev>'} ];
list [expired] {copy | backup} of
    { datafile {'<file>' | <n>} [, …]
    | tablespace '<ts>' [, …]
    | database [skip tablespace '<ts>' [, …]]
    | controlfile
    | archivelog
        { all | like '<file>' | {from | until}
                { time [=] '<date>'
                | scn [=] <n>
                | logseq [=] <n>
                  [thread = <n>]} }}
    [tag [=] <tag>] [like '<string>']
    [device type '<dev>']
    [recoverable [until
        { time [=] '<date>' | scn [=] <n>
        | logseq [=] <n> [thread [=] <n>]} ]]
    [completed { {after | before} [=] '<date>'
        | between '<date>' and '<date>'}]
    [by backup [verbose] ]
    [by {backup summary | file} ]
    [summary];
list incarnation [of database ['<id>']];
```

Backup

```
copy
    { datafile {'<file>' | <n>}
    | datafilecopy {'<file>' | tag [=] <tag>}
    | archivelog '<log>'
    | controlfilecopy {'<cntrl>' | tag [=] <tag>}
    | current controlfile }
    to '<dest>' [, …]
    [tag [=] '<tag>'] [level [=] <n>]
    [nochecksum] [check logical];
set duplex = { off | on | 1 | 2 | 3 | 4 };
```

```
backup [ full | incremental
                level [=] { 0 | 1 | 2 | 3 } ]
    [cumulative] [nochecksum]
    [check logical] [proxy [only]] [(]
    { datafile {'<file>' | <n>} [, …]
    | datafilecopy
        {'<file>' | tag [=] <tag>} [, …]
    | tablespace '<ts>' [, …]
    | database
    | archivelog
        { all | like '<log>' | {from | until}
                { time [=] '<date>'
                | scn [=] <n>
                | logseq [=] <n>
                  [thread = <n>] }}
    | current controlfile
    | controlfilecopy '<cntrl>' }
    [not backed up [since time [=] '<date>'] ]
    [plus archivelog]
    [include current controlfile]
    [delete [all] input]
    [tag [=] <tag>] [format [=] '<fm>']
    [filesperset [=] <n>] [channel <chann>]
    [skip {offline | readonly | inaccessible}]
    [setsize [=] <n>] [diskratio [=] <n>]
    [pool [=] <n>] [parms [=] '<par>'] [)];
validate backupset <n> [, …] [check logical];
```

Restore & Recovery

```
set autolocate {on | off};
set archivelog destination to '<path>';
set newname for datafile {'<file>' | <n>}
    to '<new>';
restore [(]
    { database
        [skip [forever] tablespace <ts> [, …]]
    | tablespace '<ts>' [, …]
    | datafile {'<file>' | <n>} [, …]
    | archivelog
        { all | like '<log>' | {from | until}
                { time [=] '<date>'
                | scn [=] <n>
                | logseq [=] <n>
                  [thread = <n>] }}
    | controlfile [to '<cntrl>'] } [)]
    [channel <chann>] [from tag [=] '<tag>']
    [parms '<par>']
    [from {backupset | datafilecopy} ] [validate]
    [check readonly] [check logical]
    [ until { time [=] '<date>' | scn [=] <n>
        | logseq [=] <n> [thread [=] <n>] }];
replicate controlfile from '<cntrl>';
switch datafile
    { {'<file>' | <n>} [to datafilecopy
                        {'<file>' | tag [=] <tag>} ]
    | all };
set until { time [=] '<date>' | scn [=] <n> |
    logseq [=] <n> [thread [=] <n>] };
recover
    { database
        [ until  { time [=] '<date>'
```

```
    | scn [=] <n>
    | logseq [=] <n> [thread [=] <n>] }]
    [skip [forever] tablespace <ts> [, …] ]
    | tablespace '<ts>' [, …]
    | datafile {'<file>' | <n>} [, …] }
    [delete archivelog] [check readonly]
    [check logical] [noredo];
blockrecover
    { datafile <x> block <x> [, …]
    | tablespace <ts> dba <x> [, …]
    | corruption list }
    [from {backupset | datafilecopy} ]
    [from tag [=] '<tag>']
    [restore until
        { time [=] '<date>'
        | scn [=] <n>
        | sequence [=] <n> thread [=] <n> } ];
set auxname for datafile {'<file>' | <n>}
    to {'<new>' | null };
duplicate target database
    to '<db>' [logfile
        {'<log>' [size <n>] [reuse]
        | group <n> ('<log>' [, …])
                    [size <n>] [reuse] }]
    [nofilenamecheck] [skip readonly];
```

Distributed DB, Replication, Heterogenous Services, Advanced Queuing & Data Warehousing

Views & Tables

v$dblink, v$db_pipes, vaq, vhs_agent, v$hs_session, v$hs_parameter, dba_db_links, dba_2pc_pending, dba_2pc_neighbors, dba_repcatlog, dba_repgroup, dba_repgroup_privileges, dba_repcolumn, dba_repcolumn_group, dba_repgenobjects, dba_repgrouped_column, dba_repkey_columns, dba_repsites, dba_repsites_new, dba_repobject, dba_reppriority, dba_reppriority_group, dba_repprop, dba_repddl, dba_repconflict, dba_represolution, dba_represolution_method, dba_represol_stats_control, dba_represolution_statistics, dba_repparameter_column, dba_repcat_refresh_templates, dba_repcat_template_objects, dba_repcat_template_parms, dba_repcat_template_sites, user_repcat_temp_output, dba_repcat_user_authorizations, dba_repcat_user_parm_values, dba_jobs, dba_jobs_running, deftran, dba_snapshots, snap$, dba_snapshot_refresh_times, dba_snapshot_logs, dba_snapshot_log_filter_cols, dba_registered_snapshots, dba_registered_snapshot_groups, dba_queues, dba_queue_tables, dba_queue_schedules, queue_privileges, dba_refresh, dba_refresh_children, dba_all_refresh_dependencies, dba_rchild, dba_rgroup, defcall, defcalldest, defdefaultdest, deferrcount, deferror, deflob, defpropagator, defschedule, deftran, deftrandest, dba_mviews, dba_mview_aggregates, dba_mview_joins, dba_mview_keys, dba_mview_analysis, dba_mview_detail_relations, dba_summaries, dba_summary_aggregates, dba_summary_joins, dba_summary_keys, dba_summary_detail_tables, dba_dimensions, dba_dim_levels, dba_dim_hierachies, dba_dim_child_of, dba_dim_attributes, dba_dim_join_key, dba_dim_level_key, mview$_exceptions, mviews$_recommendations, mview$_evaluations, hs_all_caps, hs_class_caps, hs_base_caps, hs_inst_caps, hs_all_dd, hs_class_dd, hs_base_dd, hs_inst_dd, hs_all_inits, hs_class_init, hs_inst_init, hs_external_objects, hs_external_object_privileges, hs_external_user_privileges, hs_fds_class, hs_fds_inst, trusted_servers

Parameters

global_names, open_links, open_links_per_instance, distributed_transactions, commit_point_strength, job_queue_processes, aq_tm_processes, dblink_encrypt_login, replication_dependency_tracking, query_rewrite_enabled, query_rewrite_integrity, hs_autoregister, hs_commit_point_strength, hs_db_domain, hs_db_internal_name, hs_db_name, hs_describe_cache_hwm, hs_language, hs_nls_date_format, hs_nls_date_language, hs_nls_nchar, hs_open_cursors, hs_rowid_cache_size, hs_rpc_fetch_reblocking, hs_fds_fetch_rows, hs_rpc_fetch_size

Packages

DBMS_REPCAT
{create | drop}_master_repgroup, {suspend | resume}_master_activity, {create | drop}_master_repobject, set_columns, {add | remove}_master_database, alter_master_propagation, relocate_masterdef, {make | drop}_column_group, {add | drop}_grouped_column, {add | drop}_update_resolution, {define | drop}_priority_group, {add | alter | drop}_priority_<type>, {alter | drop}_priority, {define | drop}_site_priority, {add | alter | drop}_site_priority_site, {add | drop}_unique_resolution, {add | drop}_delete_resolution, generate_{replication | snapshot}_support, create_snapshot_repobject, switch_snapshot_master, send_and_compare_old_values, {register | cancel | purge}_statistics, do_deferred_repcat_admin, purge_master_log, repcat_import_check, comment_on_{repgroup | repobject | repsites | column_group | priority_group | site_priority | unique_resolution | update_resolution | delete_resolution}, {specify | add}_new_masters, prepare_instantiated_master, resume_propagation_to_mdef

DBMS_REPCAT_ADMIN
grant_admin_{schema | any_schema}, register_user_repgroup

DBMS_REPCAT_INSTANTIATE

DBMS_REPCAT_RGT
create_template_object

DBMS_REPUTIL
replication_{on | off}

DBMS_DEFER
transaction, call, <type>_arg

DBMS_DEFER_SYS
{add | delete}_default_destination, push, purge, delete_tran, execute_error, execute_error_as_user, delete_error, schedule_push, unschedule_push, set_disabled, disabled, schedule_purge, schedule_execution, register_propagator

DBMS_DEFER_QUERY

DBMS_OFFLINE_OG
{begin | end}_instantiation, resume_subset_of_masters, {begin | end}_load

DBMS_OFFLINE_SNAPSHOT
{begin | end}_load

DBMS_REFRESH
refresh, change

DBMS_JOB
submit, remove, change, what, next_date, interval, broken, run, instance

DBMS_RECTIFIER_DIFF
differences, rectify

DBMS_AQ, DBMS_AQADM

DBMS_MVIEW (DBMS_SNAPSHOT)
{begin | end}_table_reorganization, purge_log, purge_direct_load_log, purge_snapshot_from_log, purge_mview_from_log, {register | unregister}_snapshot, {register | unregister}_mview, set_i_am_a_refresh, i_am_a_refresh, refresh, refresh_mv, refresh_all, refresh_all_mviews, refresh_dependent, get_log_age, get_mv_dependecies, {set | wrap}_up, testing, explain_{mview | rewrite}, pmarker

DBMS_OLAP
validate_dimension, estimate_space, recommend_mv, estimate_summary_size, evaluate_utilization, evaluate_utilization_w, set_logfile_name

DEMO_DIM
print_dim, print_alldims

DEMO_SUMADV

DBMS_HS
create_inst_init, drop_inst_init, create_fds_inst, drop_fds_inst

DBMS_HS_PASSTHROUGH
execute_immediate, open_cursor, bind_variable, execute_non_query, fetch_row, get_value, close_cursor

DBMS_DISTRIBUTED_TRUST_ADMIN
deny_all, allow_all, deny_server, allow_server

Files

catrep.sql, catdefer.sql, catrepc.sql, smdim.sql, sadvdemo.sql, caths.sql

Desupported Features

job_queue_interval, defcall, distributed_lock_timeout, snapshot_refresh_keep_connections, snapshot_refresh_processes, snapshot_refresh_interval, distributed_recovery_connection_hold_time, job_queue_keep_connections

Distributed DB, Replication, Heterogenous Services, Advanced Queuing & Data Warehousing (cont.)

Distributed DB

create [shared] [public]
    database link <link[@qual]>
    [connect to
        {<user> identified by <pwd>
        | current_user}]
    [authenticated by <user>
        identified by <pwd>]
    [using '<netserv>'];
alter session close database link <link>;
drop [public] database link <link>;
alter session advise
    {commit | rollback | nothing};
alter system {enable | disable} distributed
    recovery;
commit comment 'ORA-2PC-CRASH-TEST-
    <1-10>';

Materialized Views

create {materialized view | snapshot} log
    on <tab> [tablespace <ts>] [storage (…)]
    [pctfree <u>10</u>] [pctused <u>40</u>]
    [initrans <u>1</u>] [maxtrans <n>]
    [<u>logging</u> | nologging] [cache | <u>nocache</u>]
    [<u>noparallel</u> | parallel [<n>]]
    [partition…] [lob…] [using index…]
    [with [primary key] [, rowid]
        [(<col> [, …])]]
    [{including | <u>excluding</u>} new values];
alter {materialized view | snapshot} log
    on <tab>
    [add [primary key] [, rowid]
        [(<col> [, …])]] […];
drop {materialized view | snapshot} log
    on <tab>;
create {materialized view | snapshot} <mview>
    [tablespace <ts>] [storage (…)]
    [pctfree <u>10</u>] [pctused <u>40</u>]
    [initrans <u>1</u>] [maxtrans <n>]
    [<u>logging</u> | nologging] [cache | <u>nocache</u>]

    [<u>noparallel</u> | parallel [<n>]]
    [cluster <clust> (<col> [, …])]
    [lob…] [partition…]
    [build {<u>immediate</u> | deferred}]
    [on prebuilt table
        [{with | <u>without</u>} reduced precision]]
    [using index…]
    [refresh [fast | complete | <u>force</u>]
        [on commit | <u>on demand</u>]
        [start with '<date>'] [next '<date>']
        [with {<u>primary key</u> | rowid}]
        [using [default] [master | local]
            rollback segment [<rbs>]]]
    | never refresh]
    [for update]
    [{enable | <u>disable</u>} query rewrite]
    as <query>;
alter {materialized view | snapshot} <mview>
    … [compile];
drop {materialized view | snapshot} <mview>;

Dimensions

create [force | <u>noforce</u>]
    dimension <dim> level <lev> is [(|
    <tab>.<col> [, …])] [level…]
    hierachy <hier>
    (<child_lev> child of <parent_lev>
        [child of <parent_lev>…]
        [join key (<child_col> [, …])
            references <parent_lev>]
        [join…])
    [attribute <lev> determines
        [(| <dep_col> [, …])]] [attribute…];
alter dimension <dim>
    { add { level… | hierachy… | attribute… }
    | drop
        { level <lev> [<u>restrict</u> | cascade]
        | hierachy <hier>
        | attribute <lev> }
    | compile };
drop dimension <dim>;

Real Application Clusters

Processes

IDLM, ~~PCM~~, OPQ, OPSM, OPSD
vendor OSDs:
CM, Start, IO, IPC
(RegKeys: OSD, CMDLL, IODLL, IPCDLL, STARTDLL)

Views & Tables

gv$<dyn_perf_view>, v$active_instances, v$resource, v$resource_limit, v$ges_statistics, v$ges_latch, v$ges_convert_local, v$ges_convert_remote, v$ges_enqueue, v$ges_blocking_enqueue, v$ges_resource, v$ges_traffic_controller, v$gc_element, v$cr_block_server, v$gc_elements_with_collisions, v$cache_transfer, v$file_cache_transfer, v$temp_cache_transfer, v$class_cache_transfer, v$false_ping, v$lock_activity, v$lock_element, v$lock_class_ping, v$cache_lock, v$latch_misses, v$hvmaster_info, v$gcshvmaster_info, v$gcspfmaster_info, file_lock, ext_to_obj, oraping_config

Parameters

cluster_database, cluster_database_instances, cluster_interconnects, active_instance_count, thread, instance_name, instance_number, instance_groups, parallel_instance_group, service_names, dml_locks, gc_files_to_locks, gc_latches, max_commit_propagation_delay, parallel_default_max_scans, lock_name_space, cpu_count, trace_enabled, sessions_per_user????

Package

DBMS_LIBCACHE
compile_from_remote

Files

init<db_name>.ora, <db_name>.conf, utlclust.sql, catclust.sql, clustdb.sql, catlibc.sql, dbmslibc.sql

Desupported Features

vdlm_misc, vdlm_latch, v$dlm_convert_local, v$dlm_covert_remote, vdlm_locks, vdlm_ress, vdlm_all_locks, vdlm_traffic_controller, v$lock_element, v$bsp, v$locks_with_collisions, v$file_ping, v$temp_ping, v$ping, v$class_ping
init_com.ora

parallel_server, parallel_server_instances, ops_interconnects, gc_defer_time, gc_releasable_locks, gc_rollback_locks, lm_locks, lm_ress, gc_latches, gc_lck_procs, delayed_logging_block_cleanouts, freeze_db_for_fast_instance_recovery, ogms_home, ops_admin_group, lm_procs

SQL

alter session instance_number...??
alter {table | cluster | index} <segm>
allocate extent ([size <n>]
[datafile '<file>'] [instance <n>]);
create {table | cluster | index} <segm>
… storage (…
[freelists <1>]
[freelist groups <1>] …) …;

Utilities

srvctl
{ <cmd> } –h
| config [–p <db>]
| {start | stop} –p <db>
[–i <inst> | –s {<inst> | <lsnr>}]
| status –p <db> [–s {<inst> | <lsnr>}]
| add
{ db –p <db> –o <oracle_home>
| instance –p <db> –i <inst>
–n <node> }
| delete
{ db –p <db>
| instance –p <db> –i <inst> }
| rename instance –p <db>
–i <old> –e <new>
| move instance –p <db>
–i <inst> –n <new_node>
| get env –p <db> [–i <inst>]
| set env –p <db> –t <var> = <val>
[–i <inst>]
| unset env –p <db> –t <var> [–i <inst>] }
srvconfig
{ –init
| {–exp | –imp} <file>
| –conv <db.conf> }
gsd, gsdservice { –start | –install | –remove }
opsctl [start | stop]

–c <user>/<pwd> –n <db> [–i <sid> [, …]]
[–f] [–t] [–u] [–m] [–y | e] [–v] [–h]
GUIOracleOBJManager, setlinks /f:<file> /d
crtsrv.bat

Fail Safe & RAC Guard

fscmd
{ dumpcluster | movegroup
| onlinegroup | offlinegroup
| onlineresource | offlineresource
| verifygroup | verifyallgroups }
<resource> /cluster = <clust>
[/logfile = <log>] [/node = <node>]
[/offline =
{ abort | immediate
| transactional | normal }]
[/domain = <OSdomain>
/user = <OSuser> /pwd = <pwd>]
pfsctl
{ help | pfsboot | pfshalt
| status | restore
| move_primary [<sec>]
| stop_secondary [<sec>]
| bootone <pack> [-f]
| haltone <pack>
| switchover [<sec>]
| call_home

| report [-f <file>]
[-d yyyy/mm/ss-hh:mi:ss] [-s] }
$ORACLE_SERVICE, pfssetup
createpacks, deletepacks

Real Application Clusters (cont.)

Tuning/Contention (RAC)

Global Cache Service (GCS)

«global cache %» (v$sysstat, class 40)
contention:
    «global cache cr timeouts» = 0
    «global cache convert timeouts» = 0
cache fusion latency:
    «global cache cr block receive time» /
    «global cache cr blocks received»: ~ 15
    ms (1 ms with user mode IPC, OPS8*i*:
    ~ 1-40 ms)
    «global cache current block receive time» /
    «global cache current blocks received»
LMS service time (sum & individual):
    «global cache cr (queue + build + flush
    + send) time» / «global cache cr blocks
    served»
    «global cache current (pin + flush + send)
    time» / «global cache current blocks
    served»
average get time: ~ 20-30 ms
    «global cache get time» /
    «global cache gets»
average convert time: ~ 10-20 ms
    «global cache convert time» /

    «global cache converts»

other statistics:
    v$cache (forced_writes = 0, forced_reads),
    v$cache_transfer, v$bh, v$class_cache_
    transfer, v$file_cache_transfer, v$rowcache
    (dc_sequences, dc_used_extents)

Global Enqueue Service (GES)

«global lock %» (v$sysstat, class 32)
average global enqueue get time: ~ 20-30 ms
    «global lock get time» / («global lock sync
    gets» + «global lock async gets»)
average global lock convert time: ~ 20 ms
    «global lock convert time» / («global
    lock sync converts» + «global lock async
    converts»)

other statistics:
    v$lock_activity, v$ges_statistics, v$ges_
    convert_local, v$ges_convert_remote,
    v$rowcache, v$librarycache

Wait Events

v$system_event

statistics:
    «buffer busy %», «cr request entry», «db
    file %», «enqueue», «global cache %»,
    «KJC: wait %», «library cache pin», «log
    file sync», «row cache lock»
contention:
«global cache busy», «buffer busy due to
    global cache»

Latches

v$latch:
    gets / misses ~ 0.9-0.95
v$latch_misses:
    sleeps / misses

Sequences

use sequence number multipliers
cache sequence numbers

Tuning/Contention (OPS 8*i*)

Global cache

consistent-read requests:
    «global cache cr block received» +
    «global cache cr blocks read from disk»

Global locks

IDLM non-PCM resources:
    v$librarycache, v$rowcache

IDLM

lock statistics:
    v$dlm_convert_local, v$dlm_convert_re-
    mote
message statistics: (v$dlm_misc)
    average receive queue length: < 10
    «dlm total incoming msg queue length» /
    «dlm messages received»

OPS I/O

«DBWR forced writes» / «physical writes»
(v$sysstat)
(«remote instance undo header writes» + «re-
mote instance undo block writes») / «DBWR
forced writes» (v$sysstat)

Locking

«releasable freelist waits» (v$sysstat)

Lock conversion

lock hit ratio: (v$sysstat)
    consistent gets» – «global lock converts
    (async)» / «consistent gets» > 95%,
    «lock element cleanup»
    (v$system_event, v$session_wait),
    v$lock_activity, v$class_ping, v$ping

Pinging

ping write ratio: (v$sysstat)
    «DBWR cross instance writes» / «physical
    writes», v$lock_activity

Block contention

vbh, vcache, v$ping
    mult. copies of 2nd block of file -> freelist
    contention (check v$waitstat)

Partitioning

partition tables and indexes OR
configure process free lists and free list groups
+ allocate extents for instances (free list group
choice: «alter session set instance = <n>;»)

PCM Locks

«lm_locks» = «lm_ress» =
    2 * (gc_files_to_locks + gc_rollback_locks
    [fixed] + gc_releasable_locks),
    v$resource_limit,

Enqu. Locks

20 + (10*sess) + db_files + 1 + (2*proc) +
(db_block_buffers/64)

DML Locks

set «dml_locks» = 0 for all instances, or disable
specific table locks

Recovery

«instance recovery database freeze count»
(v$sysstat)

Inst. groups

«alter session set parallel_instance_group =
<grp>;»

Globalization Support

Views & Tables

v$nls_parameters, v$nls_valid_values,
v$timezone_names, nls_database_parameters,
nls_instance_parameters, nls_session_parame-
ters, props$, csmv$columns, csmv$constraints,
csmv$errors, csmv$indexes, csmv$tables

Packages

DBMS_SESSION
    set_nls(<name>,<value>)

Files

<prod><lang>.msb, timezone.dat,
timezlrg.dat, csminst.sql

Server: Init. Parameters

nls_language
- nls_date_language
- nls_sort

nls_territory
- nls_date_format
- nls_currency (fm L),
  nls_iso_currency (fm C),
  nls_dual_currency
- nls_numeric_characters (fm DG)
- nls_calendar
- nls_comp
- nls_length_semantics
- nls_nchar_conv_excp
- nls_time_format
- nls_timestamp_format
- nls_timestamp_tz_format
- nls_time_tz_format

$ORA_TZFILE

Client: Environment Variables

nls_lang, nls_nchar
- nls_date_language
- nls_sort
- nls_date_format
- nls_currency,
  nls_iso_currency,
  nls_dual_currency
- nls_numeric_characters
- nls_comp
- nls_calendar
- nls_credit, nls_debit

- nls_list_separator
- nls_display
- nls_monetary

Session:

alter session set nls_language = <lang>
    nls_territory = <territ>;
alter session set time_zone = <x>;

NLS-Affected SQL-Functions

to_char
- nls_date_language
- nls_numeric_characters
- nls_currency
- nls_iso_currency
- nls_calendar

to_date
- nls_date_language
- nls_calendar

to_number
- nls_numeric_characters
- nls_currency
- nls_iso_currency

nls_upper
- nls_sort

nls_lower
- nls_sort

nls_initcap
- nls_sort

nlssort
- nls_sort

Datetime Functions

Character Set & Timezone

create database …
    [character set {<charset>   << OS dependent
        | <UTF8> | <UTFE> | <AL32UTF8>}]
    [national character set   << 9i: Unicode only
        {<UTF8> | <AL16UTF16>}]
    [set time_zone =
        { '<{+|-}hh:mi>'
        | '<time_zone_region>' }] …
alter database [<db>] [national] character set
    <new_char;   << must be strict superset
    << AL24UTFFSS is desupported in 9i

Utilities

cssscan
    help = <n> userid = <user>/<pwd> parfile
    = <par> log = <scan.log> user = <schema>
    table = (<tab> [, …]) exclude = (<tab> [,
    …) tochar = <new> fromchar = <old>
    tonchar = <new> fromnchar = <old> ar-
    ray = <10240> process = <1> maxblocks
    = <x> capture = <n> suppress = <x>
    feedback = <x> boundaries = <x> lastrpt =
    <n> preserve = <n>

lbuilder

lxegen

lxinst [oranls=<$ORA_NLS33>]
    [sysdir=<path>] [destdir=<path>]
    [help=<no>] [warning={0 | 1 | 2 | 3}]

lxbcnf [oranls=<$ORA_NLS33>]
    [userbootdir=<path>] [destdir=<path>]
    [help=<no>]

Desupported Features

nls_monetary_characters, nls_list_separator,
nls_credit, nls_debit, nls_union_currency

SQL*Plus

sqlplus
    [-h[elp]
    | -v[ersion]
    | [-m[arkup]
        html [on | off] [head "<txt>"]
        [body "<txt>"] [table "<txt>"]
        [entmap {on | off}] [spool {on | off}]
        [pre[format] {on | off}]
    [-r[estrict] <1 | 2 | 3>]
    [-s[ilent]]]
    [<user>[/<pwd>][@<serv>] | /]
        [as {sysoper | sysdba} | /nolog]
    [@<URI | file>[.<ext>] [<arg>, …]]

Environment

appi[nfo] {on|off|<text>}, array[size] [<15>],
auto[commit] {on|off|imm[ediate]|<n>},
autop[rint] {on|off}, autorecovery [on|off],
autot[race] {on|off|trace[only]} [exp[lain]]
[stat[istics]], blo[ckterminator] <.>, cmds[ep]
{<;>|on|off}, colsep <_>, com[patibility]
{native|v8|v7}, con[cat] {<.>|on|off},
copyc[ommit] <0>, copytypecheck
{on|off}, def[ine] {<&>|on|off}, describe
[depth {<1>|all} | indent {on|off} | line-
num {on|off}], echo {on|off}, editf[ile]
<file>[.<ext>], emb[edded] {on|off}, esc[ape]
{<\>|on|off}, feed[back] {<6>|on|off}, flag-
ger {off|entry|intermed[iate]|full}, flu[sh]
{on|off}, hea[ding] {on|off}, heads[ep]
{||on|off}, instance {<serv>|local}, lin[esize]
<80>, lobof[fset] <1>, logsource [<path>],
long <80>, longc[hunksize] <80>, m[arkup]
html [on | off] [head "<txt>"] [body "<txt>"]
[table "<txt>"] [entmap {on | off}] [spool {on
| off}] [pre[format] {on | off}], newp[age]
{<1>|none}, null <txt>, numf[ormat] <fmt>,
num[width] <10>, pages[ize] <24>, pau[se]
{on|off|<txt>}, recsep {wr[apped]|ea[ch]|off},
recsepchar <_>, serverout[put] {on|off}
[size <2000>] [for[mat] {wra[pped]|wor[d_
wrapped|tru[ncated]}], shift[inout]
{vis[ible]|inv[isible]}, show[mode] {on|off},
sqlbl[anklines] {on|off}, sqlc[ase] {mix[ed]|lo[
wer]|up[per]}, sqlc[ontinue] <>>, sqln[umber]
{on|off}, sqlpluscompat[ibility] <x.y[.z]>,
sqlpre[fix] <#>, sqlp[rompt] <SQL>>,
sqlt[erminator] {<;>|on|off>}, suf[fix] <SQL>,
tab {on|off}, term[out] {on|off}, ti[me]
{on|off}, timi[ng] {on|off}, trim[out] {on|off},
trims[pool] {on|off}, und[erline] {<->|on|off},
ver[ify] {on|off}, wra[p] {on|off}
sql.pno, sql.lno, sql.release, sql.sqlcode, sql.user

SQL Buffer Manipulation

ed[it], a[ppend], c[hange] /<old> [/<new>],
cl[ear] buff[er], del [<n>] [<y>] [*] [last], l[ist]
[<n>] [<y>] [*] [last], i[nput]

Data Types SQL*Plus

var[iable] [<var>
    [number | char | char (<n>) | nchar
    | nchar (<n>) | varchar2 (<n>)
    | nvarchar2 (<n>) | clob | nclob
    | refcursor]]
char: max. 2.000B, varchar: max. 4000B

Commands

/, r[un]
@<file>, @@<file>, start <file>
sav[e] <file> [cre[ate] | rep[lace] | app[end]]
get <file> [lis[t] | nol[ist]]
spo[ol] [<file> | off | out]
pri[nt] [<var>], help, rem[ark], set, show
{ho[st] | ! | $} <cmd>
store [set] <file>
    [cre[ate] | rep[lace] | app[end]]
def[ine] <var> = <value>
undef[ine] <var>
pro[mpt] [<string>]
pau[se] [<string>]
conn[ect]
    { / | <user/pwd> | internal {<pwd>} }
    [as {sysdba | sysoper}]
    << desupported
disc[onnect]
passw[ord] [<user>]
startup, shutdown, recover

Data Access

attribute <object_type>.<attr>
    [ali[as] <name>] [for[mat] <fm>]
    [like <attr>] [cle[ar]] [on |off]
acc[ept] <var>
    [num[ber] | char | date]
    [for[mat] <fm>] [def[ault] <def>]
    [prompt <string> | nopr[ompt]] [hide]
desc[ribe]
    { <tab> | <view> | <pack>
    | <func> | <proc> | <syn> | <type> }
sho[w]
    { <var> | all
    | err[ors]
        [{ package | package body
        | function | procedure

| view | trigger | type | type body
| dimension | java class } <name>]
| lno | pno | user | tti[tle] | bti[tile]
| reph[eader] | repf[ooter] | spoo[l]
| sqlcode | sga | parameters | release }
timi[ng]
    [start <string> | show | stop]
exec[ute]
    { <:var> := <func> (<par> [, …])
    | <proc> (<par> [, …]) }
whenever {sqlerror | oserror}
    { exit…
    | continue [commit | rollback | none] }
{exit | quit}
    [success | failure | warning | <n>
    | <var> | <:var>] [commit | rollback]
copy [from <user>@<db>] [to <user>@<db>]
    {create | replace | insert | append}
    <tab> [(<col>, …)] using <query>;

Formatting

{ tti[tle] | bti[tle] | reph[eader] | repf[ooter] }
    [[page] [le[ft] | ce[nter] | r[ight]]
        [col <n>] [tab <n>] [bold] [s[kip] <n>]
        [format <fm>] ['<string>'] [<var>] […]
    | {on | off}]
col[umn] < col>
    { [for[mat] <fm>]
        [wra[pped] | wor[d_wrapped]
        | tru[ncated]]
        [hea[ding] <string>]
        [ali[as] <alias>] [nul[l] <string>]
        [fold_a[fter] | fold_b[efore]]
        [like <alias> | newl[ine]]
        [{new_v[alue] | old_v[alue]} <var>]
        [jus[tify] { l[eft] | c[enter] | c[entre]
            | r[ight] }]
    | {on | off} | {print | noprint} | cle[ar] }]
bre[ak]
    [on {<bcol> | row | report | <expr>}
    [ski[p] <n> | page] [on…]
    [nodup[licates] | dup[licates]]]
comp[ute]
    [{ sum | min[imum] | max[imum] | avg
    | std | var[iance] | cou[nt] | num[ber] }
    […] [la[bel] <lab>]
    of <col> [<col>…]
    on {<bcol> | row | report}]
clear
    { scr[een] | col[umns] | bre[aks]
    | comp[utes] | sql | timi[ng] | buff[er] }

Data Types (PL/SQL & Database)

Views & Tables

v$type_size, v$temporary_lobs, v$timezone_names, dba_types, dba_type_attrs, dba_type_methods, dba_coll_types, dba_lobs, dba_part_lobs, dba_lob_partitions, dba_lob_subpartitions, dba_varrays, dba_refs, dba_operators, dba_oparguments, dba_opbindings, dba_opancillary, dba_method_params, dba_method_results, dba_directories, dba_rulesets

SQL-Functions

Parameters

Scalar Types (Built-in Types)

character
- char (<1> [byte | char]) type 96
  {col: 2.000B, pl: 32.767B}
  (Subtype: character)
- varchar2 (<n> [byte | char]) type 1
  {col: 4.000B,
   pl: 32.767B (preallocated < 2000B)}
  (Subtypes: string, varchar) << deprec.
- nchar (<1>) type 96
  {col: 2.000B, pl: 32.767B, unicode only}
  national character literal:
    N'<string>'
- nvarchar2 (<n>) type 1
  {col: 4.000B, pl: 32.767B, unicode only}

binary_integer
- {pl: -2.147.483.647 .. 2.147.483.647}
  library arithmetic
  (Subtypes: natural {non-neg.}, naturaln {not null} positive{pos.}, positiven {not null}, signtype{-1,0,1})

pls_integer
- {pl: -2.147.483.647 .. 2.147.483.647}
  machine arithmetic

number [(<prec>[, <scal>])] type 2
- {precision: 38 digits, scale: -84 to 127, 21B (20B Mantisse, 1B Exponent)}
  (Subtypes: dec, decimal, double precision, float, int, integer, numeric, real, smallint)

datetime and interval
- date   type 12/13
  {7B = CentYearMonDayHourMinSec, 8B, -4.712 to 9.999}
  ANSI date literal:
    date '<yyyy-mm-dd>'
- timestamp [(<6>)]   type 180
  [with time zone   type 181/187/188
  | with local time zone]   type 231
  {20B}
  timestamp literal:
    timestamp '<yyyy-mm-dd ...>'
- interval year [(<2>)] to month   type 182
- interval day [(<2>)] to second [(<6>)]
  type 183

raw (<n>) type 23
- {col: 2.000B, pl: 32.767B}

large objects

- long type 8   << deprecated
  {col: 2³¹-1B=2G, pl: 32.760B}
- long raw type 24   << deprecated
  {col: 2³¹-1B=2G, pl: 32.760B}
- internal:
  CLOB, NCLOB type 112
  BLOB type 113
  {col: 2³²-1B=4G,
  inline ~4000b, else out of line}
- external:
  BFILE {pointer} type 114
  {ext. LOB: 2³²-1B=4G}
  create [or replace] directory <dir> as '<path>';
  drop directory <dir>;

rowid
- rowid   type 69
  {extended: 10B,
   restricted: 6B (block.row.file),
   physical rowid}
- urowid [(<4000B>)]   type 208
  col: 4.000B (IOT logical urowid or foreign table foreign urowid)

boolean
- {pl: true | false | null}

subtype <subtype> is <base_type> [not null];

ANSI Supported Types

character [varying] (<n>)
{char | nchar} varying (<n>)
varchar (<n>)
national {character | char} [varying] (<n>)
{numeric | decimal | dec} [(<prec>[, <scal>])]
{integer | int | smallint}
float [(<n>)]
double precision
real

Relationship Types

ref
- ref cursor, ref <otype>
  {pointer}

Record Types

logical unit of dissimilar types
record may not be DB col
type <rec_type> is record
   (<field> {<type> | <tab>.<col>%type}
   [[not null] {:= | default} <expr>] [, …]);
<record> {<rec_type> | <tab>%rowtype};
<rec_var>.<field> := <expr>;

Collection

elements of same type
initialized by constructor <collect>(…)

varray	may be DB col
nested table	may be DB col
index-by table	must not be DB col

type <varr_type> is
   {varray | varying array} (<size>) of <type>
   [not null];

type <tab_type> is table of <type> [not null]
   [index by binary_integer];

<coll> {<varr_type> | <tab_type>};
<coll>(<subscript>)[.<item>] := <expr>;
<coll>.<method>
   count, delete [(<i> [,<j>])], exists(<i>),
   extend [(<n> [,<i>])], limit, first, last,
   next(<i>), prior(<i>), trim [(<i>)]

User-defined Types

abstract types
initialized by constructor <type>(…)
create [or replace] type <type>;
   forward type definition / incomplete type
create [or replace] type <type>
   [authid {current_user | definer}] {is | as}
   { object (<attr> <type> [, …]
      [, {static | [map | order] member}
      {function | procedure} <func>
      [({self | <par>} [in | out | in out]
         <type> [, …])] [return <type>]
      [{is | as} language
            { java name '<func>'
            | C [name <func>] library <lib>
            [with context]
            [parameters (<par>)] }]
      [, pragma restrict_references
      ({<method> | default},
        {rnds | wnds | rnps | wnps | trust})]
      [, …]])
   | {varray | varying array} (<n>) of <type>
   | table of <type> };
create [or replace] type body <type>
   {is | as} {static | [map | order] member}
   {function | procedure} <func>
   [({self | <par>} [in | out | in out]
      <type> [, …])] [return <type>]
   [{is | as}
      { begin <stat>; end [<func>];
      | language
            { java name '<func>'
            | C [name <func>]
            library <lib> [with context]
            [parameters (<par>)] }}
   [, …] end;
alter type <type>
   { compile [debug] [specification | body]
   | replace as object (<attr> <type> [, …]
      [, {static | [map | order] member}
      {function | procedure} <func>
      [({self | <par>} [in | out | in out]
         <type> [, …])] [return <type>]
      [, pragma restrict_references
      ({<method> | default},
        {rnds | wnds | rnps | wnps | trust})]
      [, …]])};
drop type [body] <type> [force];
[ref] obj_type, type, varchar2(x), number[(p,s)], date, raw(x), char[acter](x), char varying(x), varchar(x), numeric[(p,s)], dec[imal] [(p,s)], int[eger], smallint, float[(x)], double precision, real, blob, clob, bfile

Maximum Row Size

row header (min. 3B) + SUM(max. field length + length indicator (<=250: 1B, >250: 3B))

Data Types (PL/SQL & Database) (cont.)

Oracle Supplied Types

SYS.AnyData
SYS.AnyType
SYS.AnyDataSet
SYS.XMLType
SYS.UriType
SYS.UriFactoryType
MDSYS.SDO_Geometry
ORDSYS.ORDAudio
ORDSYS.ORDImage
ORDSYS.ORDVideo

Explicit Type Conversion (Cast Function)

cast ({ <expr> | (<subquery>) | multiset (<subquery>) } as <type>)

	char, varchar2	number	datetime, interval	raw	rowid, urowid	nchar, nvarchar2
char, varchar2	X	X	X	X	X	
number	X	X				
date, timestamp, interval	X		X			
raw	X			X		
rowid, urowid	X				X	
nchar, nvarchar2		X	X	X	X	X

Explicit Type Conversion (SQL Conversion Functions)

	char, varchar2, nchar, nvarchar2	number	datetime/interval	raw	rowid	long, long raw	clob, nclob, blob
char, varchar2, nchar, nvarchar2	to_char (char), to_nchar (char)	to_number	to_date, to_timestamp, to_timestamp_tz, to_yminterval, to_dsinterval	hextoraw	chartorowid		to_clob, to_nclob
number	to_char (number), to_nchar (number)	—	to_date, to_yminterval, to_dsinterval				
datetime/interval	o_char (date), to_nchar (datetime)		—				
raw	rawtohex, rawtonhex			—			to_blob
rowid	rowidtochar				—		
long, long raw						—	to_lob
clob, nclob, blob	to_char, to_nchar						to_clob, to_nclob

Implicit Type Conversion

	char	varchar2	date	datetime/ interval	long	number	raw	rowid	clob	blob	nchar	nvarchar2	nclob
char	—	X	X	X	X	X	X		X		X	X	
varchar2	X	—	X	X	X	X	X	X	X		X	X	
date	X	X	—								X	X	
datetime/ interval	X	X		—	X		X				X	X	
long	X	X		X	—	X			X		X	X	X
number	X	X				—					X	X	
raw	X	X			X		—			X	X	X	
rowid	X	X						—			X	X	
clob	X	X			X				—				
blob							X			—			
nchar	X	X	X	X	X	X	X	X			—	X	X
nvarchar2	X	X	X	X	X	X	X	X			X	—	X
nclob					X						X	X	—

SQL, PL/SQL & Java

Views & Tables

v$reserved_words, v$resumable, dba_source, dba_errors, dba_dependencies, deptree, ideptree, dba_libraries, dba_outlines, dba_outline_hints, dba_resumable, outln.ol$, outln.ol$hints, java$options, java$class$md5$table, create$java$lob$table, dba_workspace_sessions, all_workspaces, all_workspace_privs, all_workspace_savepoints, all_version_hview, all_wm_locked_tables, all_wm_modified_tables, all_wm_ric_info, all_wm_tab_triggers, all_wm_versioned_tables, role_wm_privs, user_wm_privs, <tab>_conf, <tab>_diff, <tab>_lock, <tab>_lt, <tab>_hist, <tab>_wm

Parameters

optimizer_mode, db_file_multiblock_read_count, optimizer_features_enable, optimizer_index_caching, optimizer_index_cost_adj, optimizer_max_permutations, complex_view_merging, partition_view_enabled, hash_join_enabled, hash_area_size, star_transformation_enabled, row_locking, sql_trace, timed_statistics, create_stored_outlines, use_stored_outlines, utl_file_dir, plsql_v2_compatibility, remote_dependencies_mode, undo_retention, plsql_compiler_flags, cursor_sharing

Packages

DBMS_STANDARD,
DBMS_TRACE
    {set | pause | resume | clear | comment | limit}_plsql_trace, plsql_trace_version, get_plsql_trace_runnumber, internal_version_check
DBMS_LOCK
DBMS_DESCRIBE
DBMS_METADATA
    get_{ddl | xml | dependent_ddl | dependent_xml | granted_ddl | granted_xml | query}, open, fetch_{ddl | ddl_text | xml | clob}, close, set_{filter | count | parse_item | debug}, add_transform, set_transform_param, free_context_entry
DBMS_FLASHBACK
    enable_at_{time | system_change_number}, disable, get_system_change_number
DBMS_RESUMABLE
    abort, {get | set}_timeout, {get | set}_session_timeout, space_error_info
DBMS_DDL
DBMS_DEBUG
DBMS_PROFILER
DBMS_ALERT
DBMS_OUTPUT
    put, {new | put | get}_line, get_lines, enable, disable
DBMS_PIPE

{pack | unpack}_message[_{raw | rowid}], next_item_type, {send | receive}_message, unique_name_session, purge
DBMS_SQL
    {open | close}_cursor, parse, last_error_position, bind_{variable | array}, define_{column | column_long | array}, execute, describe_columns, fetch_rows, execute_and_fetch, last_row_{count | id}, {column | variable}_value, column_value_long, is_open, last_sql_function_code
DBMS_LDAP
DBMS_TRANSACTION
    advise_{commit | nothing | rollback}, commit, commit_{comment | force}, local_transaction_id, purge_lost_db_entry, purge_mixed, read_{only | write}, rollback, rollback_{force | savepoint}, savepoint, step_id, use_rollback_segment, begin_discrete_transaction
>> Discrete transactions do not generate undo information! <<
DBMS_WM
    {alter | goto}savepoint, {create | alter | merge | compress | freeze | get | goto | refresh | remove | rollback | unfreeze}workspace, {begin | commit}resolve, {get | set}conflictworkspace, {remove | compress}workspacetree, {enable | disable}versioning, {get | set}diffversions, getlockmode, {get | set}multiworkspaces, getopcontext, {grant | revoke}{system | workspace}priv, getprivs, gotodate, copyforupdate, {create | delete}savepoint, isworkspaceoccupied, {lock | unlock}rows, {merge | refresh | rollback}table, resolveconflicts, rollback{resolve | tosp}, set{locking | wooverwrite | workspacelockmode}{on | off}
DBMS_JAVA
    server_{startup | shutdown}, longname, shortname, {get | set | reset}_compiler_option, set_{output | streams}, {start | end}_{import | export}, {start | stop}_debugging, register_endpoint, notifiy_at_{startup | shutdown}, remove_from_{startup | shutdown}
DBMS_LOB
    append, close, compare, converttoclob, copy, {create | free}temporary, erase, fileclose, filecloseall, fileexists, filegetname, fileisopen, fileopen, getchunksize, getlength, instr, isopen, istemporary, loadfromfile, open, read, substr, trim, write, writeappend
DBMS_OBFUSCATION_TOOLKIT
    desencrypt, desdecrypt, des3encrypt, des3decrypt, md5, desgetkey, des3getkey
UTL_FILE
    fopen, fopen_nchar, is_open, fclose, fclose_all, fflush, new_line, get_line,

get_line_nchar, put_line, put_line_nchar, put, put_nchar, putf, putf_nchar, fcopy, fgetattr, fgetpos, fremove, frename, fseek, get_raw, put_raw
UTL_HTTP
UTL_URL
UTL_TCP
UTL_SMTP
UTL_ENCODE
UTL_INADDR
UTL_RAW
    cast_{to | from}_{number | binary_integer}
OUTLN_PKG
    drop_unused, {drop | update}_by_cat, drop_{collision | extras | unrefd_hints}[_expact], deptree_fill
SQLJUTL
    has_default

Files

tracetab.sql, utldtree.sql, initjvm.sql, utljavarm.sql, sqljutl.sql, owminst.plb

Desupported Features

hash_multiblock_io_count, optimizer_percent_parallel, always_anti_join, always_semi_join, fast_full_scan_enabled, push_join_predicate

SQL, PL/SQL & Java (cont.)

Number Functions

abs, acos, asin, atan, atan2, bitand, ceil, cos, cosh, exp, floor, ln, log, mod, power, round, sign, sin, sinh, sqrt, tan, tanh, trunc, width_bucket

Character Functions

chr, concat, initcap, lower, lpad, ltrim, nls_initcap, nls_lower, nlssort, nls_upper, replace, rpad, rtrim, soundex, substr[b|c|2|4], translate, treat, trim, upper
ascii, instr[b|c|2|4], length[b|c|2|4]

Datetime Functions

add_months, current_date, current_time-stamp, dbtimezone, extract, from_tz, last_day, localtimestamp, month_between, new_time, next_day, numtodsinterval, numtoyminterval, round, sessiontimezone, sys_extract_utc, systimestamp, sysdate, to_dsinterval, to_time-stamp, to_timestamp_tz, to_yminterval, trunc, tz_offset

Conversion Functions

asciistr, bin_to_num, cast…[multiset], chartorowid, compose, convert, decompose, hextoraw, numtodsinterval, numtoyminterval, rawtohex, rawtonhex, rowidtochar, rowidton-char, to_char, to_clob, to_date, to_dsinterval, to_lob, to_multi_byte, to_nchar, to_nclob, to_number, to_single_byte, to_yminterval, translate…using, unistr

Micellaneous Functions

bfilename, coalesce, decode, dump, empty_blob, empty_clob, existsnode, extract, greatest, least, nls_charset_decl_len, nls_charset_id, nls_charset_name, nullif, nvl, nvl2, sys_con-nect_by_path, sys_context, sys_dburigen, sys_extract_utc, sys_guid, sys_typeid, sys_xm-lagg, sys_xmlgen, uid, user, userenv, vsize

Aggregate Functions

avg, corr, count, covar_pop, covar_samp, cume_dist, dense_rank, first, group_id, group-ing, grouping_id, last, max, min, percen-tile_cont, percentile_disc, percent_rank, rank, regr, stddev, stddev_pop, stddev_samp, sum, var_pop, var_samp, variance

Object Reference Functions

deref, make_ref, ref, reftohex, value

Format Models

<fm> = 9 0 $ B MI S PR D G C L , . V
    EEEE RN DATE A<n>

Analytic Functions

Ranking:
    { rank() | dense_rank() | cume_dist()
    | percent_rank() | ntile(<n>)
    | row_number() }
    over ([partition by <col> [, …]] order by
    <col> [, …] [asc | desc] [nulls {first | last}])
Window Aggregate:
    { count | sum | avg | min | max | stddev
    | variance | var_samp | var_pop
    | stddev_samp | stddev_pop | covar_samp
    | covar_pop | regr_slope | regr_intercept
    | regr_r2 | regr_avgx | regr_avgy
    | regr_count | regr_sxx | regr_sxy
    | regr_syy }
    (<col>) over (
        [partition by <col> [, …]]
        order by <col> [, …] {rows | range}
        [{ between <n> | unbounded
            | interval '<n>' day } preceding]
        [{ [and] <n> | unbounded
            | interval '<n>' day } following]
        [current row]
        [first_value()] [last_value()]
        [asc | desc] [nulls {first | last}])
Reporting Aggregate:
    {<WA-Func> | ratio_to_report}
    (<col>) over (
        [partition by <col> [, …]]
        [asc | desc] [nulls {first | last}])
LAG/LEAD:
    {lag | lead} (<col>, <default>) over (
        order by <col> [, …] [asc | desc]
        [nulls {first | last}])

SQL Statement Types

DDL
    create, alter, drop, truncate, rename,
    comment, grant, revoke, audit, noaudit,
    analyze, {associate | disassociate} statistics
DML
    select, insert, update, delete, merge, lock
    table, explain plan, call
TxCtl
    commit, rollback, savepoint, set transac-tion
SessCtl
    alter session, set role
SysCtl
    alter system

Optimizer

Access Paths
1 single row by rowid
2 single row by cluster join
3 single row by hash cluster key with unique or primary key
4 single row by unique or primary key
5 cluster join
6 hash cluster key
7 indexed cluster key
8 composite key
9 single-column indexes
10 bounded range search on indexed columns
11 unbounded range search on indexed columns
12 sort-merge join
13 max or min of indexed column
14 order by on indexed columns
15 full table scan
-- sample table scan
-- fast full index scan
-- index join
-- bitmap index scan

Hints

{select | update | delete}
    { /*+ <HINT> [text] */
    | --+ <HINT> [text] }

RULE, CHOOSE, ALL_ROWS, FIRST_ROWS [(<n>)], FULL (<tab>), ROWID (<tab>), CLUSTER (<tab>), HASH (<tab>), HASH_AJ, HASH_SJ, INDEX (<tab> [<ind> […]]), INDEX_ASC (<tab> [<ind> […]]), INDEX_DESC (<tab> [<ind> […]]), INDEX_COMBINE (<tab> [<ind> […]]), INDEX_JOIN (<tab> [<ind> […]]), INDEX_FFS (<tab> [<ind> […]]), NO_INDEX (<tab> [<ind> […]]), MERGE_AJ, MERGE_SJ, AND_EQUAL (<tab> <ind> <ind> […]), USE_CONCAT, NO_EXPAND, NOREWRITE, REWRITE [(<mview> [, …])], ORDERED, STAR, USE_NL (<tab> […]), USE_MERGE (<tab> […]), USE_HASH (<tab> […]), DRIVING_SITE (<tab> […]), PARALLEL (<tab> [, {<n> | default} { [<n> | default}]]), NOPARALLEL (<tab> […]), PQ_DIS-TRIBUTE (<tab> [,] <out>, <in>), APPEND, NOAPPEND, PARALLEL_INDEX (<tab> [<ind> [, …]] [, {<n> | default} [{<n> | default}]]), NOPARALLEL_INDEX (<tab> [<ind> [, …]]), CACHE (<tab> […]), NOCACHE (<tab> […]), MERGE (<tab>), NOMERGE (<tab>), PUSH_JOIN_PRED (<tab>), NO_PUSH_JOIN_PRED (<tab>), PUSH_SUBQ, STAR_TRANSFORMA-TION, ORDERED_PREDICATES, CUR-SOR_SHARING_EXACT, DYNAMIC_SAMPLING ([<tab>] <n>)

Serial direct-load insert:
    insert /*+APPEND */ into <tab> <query>;
Parallel direct-load insert:
    alter session {enable | force} parallel dml;
    insert /*+PARALLEL(<tab>,<n>) */ into
        <tab> <query>;

SQL, PL/SQL & Java (cont.)

Queries
select
    { [aggr_func ()
        [{distinct | unique} | all]
        { [<alias>.]<col> | * }])]
        [{ + | - | * | / } <expr>]
        [as] [“<alias>”] [, …]
    | <seq>.{nextval | currval}
    | cursor (<subquery>) }
from
    { [(| [<schema>.]
        <tab/view/snapshot>[@<dblink>]
        [partition (<part>)]
        [subpartition (<subpart>)]
        [<alias>] [, …] [sample [block] (<n>)]
        [[inner | {left | right | full} [outer]]
          join <tab> { on <expr>
                | using (<col> [, …])) }
        | cross join <tab>
        | natural [inner | {left | right | full}
             [outer]] join <tab>]])]
    | (<subquery> [with {read only | check
        option [constraint <constr>]}])
    | table (<coll_expr>) [(+)] }
where
    [(| [(| [<alias>.]<col/expr> [(+)]
           [,<expr>…)]]
    { { = | != | ~= | <> | <= | >= | < | > }
        [any | some | all]
        (<expr> [, …] | <subquery>)
    | like ‘[_%]<string>‘
    | [not] in (<expr> [, …] | <subquery>)
    | [not] between <expr> and <expr>
    | = [<alias>.]<col>
    | [not] exists (<subquery>)
    | is [not] null
    | is dangling }
    [{and [not] | or} <expr>] [, …]])]
    [[start with <expr>]
        connect by [prior] <expr>]
    [group by [{rollup | cube} (]
        <expr> [, …]])]
        [having <expr> | (<subquery>)]]]
    [{union [all] | intersect | minus}
        (<subquery>)]
    [order by {<col> | <n>} [asc | desc] [, …]]
    [for update [of <tab>.<col>]
        [nowait | wait <n>]];
with <query_name> as (<subquery>) [, …]
    select … from <query_name> … ;
      << subquery factoring

DML
insert into
    { <tab> [partition (<part>)]
    | [the] <subquery1> }
    [(<col> [, …])]
    { values (<expr>, …)
    | <subquery2> } [ref into <item>];
insert
    { all into <tab>… [, <tab>…]
    | [all | first]
        when <expr> then into <tab>…
        [else into <tab>…] }
    <subquery>;

update <tab>
    set <col> = {<val> | ‘<string>’} [, …];
merge into <tab1>
    using <tab2> on (<join_expr>)
    when matched then update set…
    when not matched then insert
        (<col>…) values (…);
delete [from] <tab>
    [partition (<part>)] [alias]
    [where <expr>];

Control
commit [work]
    [comment ‘<string>’
    | force ‘<id>’ [,<scn>]];
savepoint <sp>;
rollback [work]
    [to [savepoint] <sp> | force ‘<id>’];
set transaction
    { read only | read write
    | isolation level
        {serializable | read committed}
    | use rollback segment <rbs>}
    [name ‘<tx>’];
alter session {enable | disable}
    commit in procedure;
alter session {enable | disable | force}
    parallel {dml | ddl} [parallel <n>];
alter session {enable | disable}
    resumable [timeout <7200>]
        [name ‘<str>’];

PL/SQL
declare
    { -- <comment>
    | /* <comment> */ }
    pragma autonomous_transaction;
    pragma serially_reusable;
    pragma restrict_references
        (<name>, rnds, wnds, rnps,
        wnps, trust);
    read, write, no, database, package, state
    pragma exception_init (<exc>, <err_no>);
    <var> [constant]
        { <type> | <tab>.<col>%TYPE
        | <var>%TYPE | <tab>%ROWTYPE }
        [[not null] { := | default } <n>];
    cursor <curs>
        [(<par> <type> [, …])] is
        <query> [for update of <col> [, …]];
    type <refcurs_type> is
        ref cursor return <type>;
    <refcurs> <refcurs_type>;
    type <rec_type> ist record (<col> [, …]);
    <rec> <rec_type>;
    <exc> exception;
begin [<< <blocklabel> >>]
    select …
        [[bulk collect] into <var> [, …]]
        from …;
    insert into
        {<tab> | table (<subquery>)}
            [(<col> [, …])]
        {values (<expr>, …) | <subquery>}

    [returning <expr> [, …]
    [bulk collect] into <var> [, …]];
update
    {<tab> | table (<subquery>)}
    set <col> = <expr> [, …]
    [where {<expr> | current of <curs>}]
    [returning <expr> [, …]
        [bulk collect] into <var> [, …]];
delete from
    {<tab> | table (<subquery>)}
    [where {<expr> | current of <curs>}]
    [returning <expr> [, …]
        [bulk collect] into <var> [, …]];
execute immediate <‘dyn_sql_stat>
    [[bulk collect] into
        {<var> [, …] | <rec>}]
    [using [in | out | in out] <arg> [, …]]
    [{returning | return} [bulk collect]
        into <arg> [, …]];
open <refcurs>
    for <‘dyn_multi_row_query’>
    [using <var> [, …]];
open <curs> [(<par>, …)];
<curs>%{ found | isopen | notfound | rowcount }
fetch <curs> [bulk collect] into
    {<var> [, …] | <rec>} [limit <n>];
close <curs>;
if <expr> then <stat>
    [elsif <expr> then <stat>;]
    [else <stat>;]
    end if;
[<< <label> >>]
    [while <expr>
    | for <i> in [reverse] <a>..<b>
    | for <rec> in <curs> [(<par>, …)]
    | (<query>) }]
    loop <stat>;
    [if <expr> then exit;]
    [exit [<label>] when <expr>;]
    end loop [<label>];
forall <i> in <a>..<b> [save exceptions]
    { <stat> [returning <col>
        bulk collect into <collect>];
    | execute immediate
        <upd | ins | del> … }
SQL%{bulk_rowcount(i)
| bulk_exceptions(i).error_{index | code}
| bulk_exceptions.count }
lock table <tab>
    in {share [row exclusive] | exclusive}
    mode [nowait];
set transaction
    {read only | read write
    | isolation level
        {serializable | read committed}
    | use rollback segment <rbs>};
commit [work] [comment ,<str>’];
savepoint <sp>;
rollback [work] [to [savepoint] <sp>];
null;
[<< <label> >>]
goto <label>;
{<var> := <func> | <proc>}
    ([<form_par> =>] <act_par> [, …]);
return [[(| <expr> [)]];
raise <exc>;

SQL, PL/SQL & Java (cont.)

```
exception
    when {<exc> | others} [or <exc2> …]
    then <stat>; [sqlcode; sqlerrm(<n>);]
    raise;
    >> Predefined Server Exceptions:
    no_data_found, too_many_rows, invalid_cursor,
    zero_divide, dup_val_on_index <<
end;
```

Packages

```
create [or replace] package <pack>
    [authid {current_user | definer}]
    {is | as} {procedure | function}
    <name> (<par> <type> [, …])
    [return <type>];
    [type <refcurs> is ref cursor
        return <type>;]
    end [<pack>];
create [or replace] package body
    <pack> {is | as}
    {procedure | function} <name>
    [(<par> [ in | out [nocopy]
        | in out [nocopy] ] <type>
        [{:= | default} <expr>] [, …] )]
    [return <type>]
    [authid {current_user | definer}]
    { is begin <stat>; end; end [<pack>];
    | is external library <lib>
        [name '<func>'] [language <lang>]
        [calling standard {C | pascal}]
        [with context]
    | as [language <lang>] name
        '<func> (<par>, …) return <type>' };
drop package [body] <pack>;
alter package <pack> compile
    [debug] [package | specification | body];
```

Procedures & Functions

```
create [or replace] function <func>
    [(<par> [ in | out [nocopy]
        | in out [nocopy] ] <type>
        [{:= | default} <expr>] [, …] )]
    return <type>
    [authid {current_user | definer}]
    [deterministic] [parallel_enable]
```

```
    { is <var> <type>;
        begin <stat>;
        end [<func>];
    | as external library <lib>
        [name '<func>'] [language <lang>]
        [calling standard {C | pascal}]
        parameters (
            { <par> [ indicator | length | maxlen
                | charsetid | charsetform ]
                [by ref] <type> [, …]
                return [indicator | …]
                [by ref] <type>
            | context } [, …]) [with context]
    | as [language <lang>] name
        '<func> (<par>, …) return <type>' };
drop function <func>;
create [or replace] procedure <proc>
    [(<par> [ in | out [nocopy]
        | in out [nocopy] ] <type>
        [{:= | default} <expr>] [, …] )]
    [authid {current_user | definer}]
    { is <var> <type>;
        begin <stat>;
        end [<proc>];
    | as [language <lang>] name
        '<func>(<par>, …)' };
drop procedure <proc>;
alter {function | procedure} <name>
    compile [debug];
call {<proc> | <func> | <method>}[@<dblink>]
    (<expr> [, …])
    [into <:var> [indicator <:ind>]];
```

Java

```
create [or replace]
    [and {resolve | compile} [noforce] java
    { {source | resource} named "<java>"
    | class [schema <schema>] }
    [authid {current_user | definer}]
    [resolver (
        ("<pack/class | * >" [,] {<schema> | - } )
        […] )]
    { as <src_text>
    | using { bfile (<dir>, '<file>')
```

```
    | {clob | blob | bfile} <subquery>
    | '<key_for_BLOB>'} };
alter java {source | class} "<java>"
    [resolver…]
    { {compile | resolve}
    | authid {current_user | definer} };
drop java {source | class | resource} "<java>";
```

Miscellaneous

```
create [or replace] library <lib> {is | as} '<file>';
drop library <lib>;
create [or replace] operator <oper>
    binding (<type> [, …]) return <type>
    [ancillary to <prim> (<type> [, …] )]
    [with index context] [scan context]
    [compute ancillary data] using <func>;
create [or replace] indextype <itype>
    for <oper> (<par> [, …]) using <package>;
create [or replace] context <namespace>
    using <pack|type>;
drop context <namespace>;
create schema authorization <schema>
    {create table… | create view… | grant…};
explain plan [set statement_id = '<string>']
    [into <tab>] for <stat>;
create [or replace] outline <outln>
    [for category <cat>] on <stat>;
alter outline <outln>
    { rebuild | rename to <new>
    | change category to <newcat> };
drop outline <outln>;
```

Boolean Conditions

AND	true	false	null	OR	true	false	null	NOT	
true	true	false	null	**true**	true	true	true	**true**	false
false	false	false	false	**false**	true	false	null	**false**	true
null	null	false	null	**null**	true	null	null	**null**	not null

Embedded SQL

```
exec oracle define <symbol>;
exec oracle {ifdef | ifndef} <symbol>;
exec oracle {else | endif};
exec oracle option (<name> = <value>);
exec sql include {oraca | sqlca};
sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc
exec sql enable threads;
exec sql declare <db> database;
exec sql connect
    {:<user> identified by <:pw> | <:user_pw>}
    [ at <db> using <:db>]
    { in {sysdba | sysoper} mode]
    | [alter authorization <:new_pw>] };
exec sql whenever
    {not found | sqlerror | sqlwarning}
    { continue | goto <label> | stop
    | do {<routine> | break | continue} };
exec sql declare <tab> table
    (<col> <type> [not null] [, …]);
exec sql declare <tab> table of <obj_type>;
exec sql declare <type> type as
    { object ( <col> <type> [, …])
    | varray (<size>) of <elem_type>
    | table of <obj_type> };
exec sql type <typ> is <datatype> [reference];
exec sql call <proc>(<par> [, …] )
    [into <:var> [[indicator] <:ind>] [, …]];
exec sql register connect using <:ext_proc>
    [{return | returning} <:cont>];
exec sql var <:var> is <type>
    { [(<len> | <prec>,<scal>)]
        [convbufsz [is] (<n>)]
    | [convbufsz [is] (<n>)] };
exec sql [at <db>] allocate <:curs_var>
    [[indicator] <:ind>];
exec sql [at <db>] commit [work]
    [ [comment '<str>'] [release]
    | force '<id>' [,<n>] ];
exec sql [at <db>] savepoint <sp>;
exec sql [at <db>] rollback [work]
    [ to [savepoint] <sp>
    | force '<id>' | release ];
```

Static SQL

```
exec sql [at <db>] select <val> into <:var>…
    from <tab> where <expr>…;
exec sql [at <db>] [for <n>]
    insert into {<tab> | (<subquery1>)}
    [(<col> [, …] ]
    {values (<expr> [, …]) | <subquery2>}
    [{return | returning} <expr> [, …]
        into <:var> [[indicator] <:ind>] [, …]];
exec sql [at <db>] [for <n>] update <tab>
```

```
set <col> = <expr>
    [where {<expr> | current of <curs>}]
    [{return | returning} <expr> [, …]
        into <:var> [[indicator] <:ind>] [, …]];
exec sql [at <db>] [for <n>] delete [from]
    {(<subquery>) | <tab>} [alias]
    [where {<expr> | current of <curs>}]
    [{return | returning} <expr> [, …]
        into <:var> [[indicator] <:ind>] [, …]];
exec sql [at <db>] execute
    begin <stat>; [, …] end;
    end-exec;
exec sql [at <db>] declare <curs>
    for <static_stat>;
exec sql open <curs> [using <:var>];
exec sql fetch <curs> into <:var> [, …];
exec sql close <curs>;
```

Oracle dyn. SQL method 1

non-query, no bind vars
```
exec sql [at <db>] execute immediate
    {<:str> | '<str>'};
```
<str> may be PL/SQL block

Oracle dyn. SQL method 2

non-query, known number and types of bind vars
```
[exec sql at <db>] declare <stat> statement;]
exec sql prepare <stat> from {<:str> | '<str>'};
exec sql execute <stat> [using <:var> [, …] ];
```

Oracle dyn. SQL method 3

query, known number of columns and known number
and types of bind vars
```
[exec sql at <db>] declare <stat> statement;]
exec sql [at <db>] prepare <stat>
    from {<:str> | ,'<str>'};
exec sql declare <curs> cursor for <stat>;
exec sql [for <n>] open <curs>
    [using <:var> [[indicator] <:ind>] [, …]];
exec sql [for <n>] fetch <curs> into <:var>
    [[indicator] <:ind>] [, …];
exec sql close <curs>;
```

Oracle dyn. SQL method 4

query, unknown number of columns or unknown
number or types of bind vars
```
[exec sql at <db>] declare <stat> statement;]
exec sql prepare <stat> from {<:str> | ,'<str>'};
exec sql [for <n>] execute <stat>
    [using descriptor <bind_descr>];
exec sql [at <db>] declare <curs> cursor
    for <stat>;
exec sql describe bind variables for <stat>
```

```
    into <bind_descr>;
exec sql [for <n>] open <curs>
    [using descriptor <bind_descr>];
exec sql describe [select list for] <stat>
    into <sel_descr>;
exec sql [for <n>] fetch <curs>
    using descriptor <sel_descr>;
exec sql close <curs>;
```

ANSI dyn. SQL method 4

```
exec sql [for <n>] allocate descriptor
    [global | local] {<:descr> | '<descr>'}
    [with max <100>];
exec sql prepare <stat> from {<:str> | '<str>'};
exec sql describe input <stat>
    using [sql] descriptor
    [global | local] {<:descr> | '<descr>'};
exec sql [for <n>] set descriptor
    [global | local] {<:descr> | '<descr>'}
    { count = <n> | value <item_no>
        { type | length | [ref] indicator
        | [ref] data | character_set_name
        | [ref] returned_length
        | national_character
        | host_stride_length
        | indicator_stride_length
        | returned_length_stride
        | user_defined_type_{name | name_
            length | schema | schema_length }}
        = <:var> [, …] };
exec sql [for <n>] execute <stat>
    [using [sql] descriptor
        [global | local] {<:descr> | '<descr>'}]
    [into [sql] descriptor
        [global | local] {<:descr> | '<descr>'}];
exec sql execute immediate {<:str> | '<str>'};
str> may be PL/SQL block
exec sql [at <db>] declare
    <curs> cursor for <stat>;
exec sql [for <n>] open <curs>
    [using [sql] descriptor
        [global | local] {<:descr> | '<descr>'}]
    [into [sql] descriptor
        [global | local] {<:descr> | '<descr>'}];
exec sql describe output <stat>
    using [sql] descriptor
        [global | local] {<:descr> | '<descr>'};
exec sql [for <n>] fetch <curs>
    into [sql] descriptor
        [global | local] {<:descr> | '<descr>'} };
exec sql [for <n>] get descriptor
    [global | local] {<:descr> | '<descr>'}
    { <:var> = count
```

Embedded SQL (cont.)

| value <item_no> <:var> =
    { type | length | octet_length
    | returned_octet_length | precision
    | scale | nullable | name
    | character_set_name | indicator | data
    | returned_length | national_character
    | internal_length | host_stride_length
    | indicator_stride_length
    | returned_length_stride
    | user_defined_type_{version | name
          | name_length | schema
          | schema_length}
    } [, …] };
exec sql close <curs>;
exec sql deallocate descriptor
    [global | local] {<:descr> | '<descr>'};

Collections

exec sql [at <db>] collection describe
    <:coll> [[indicator] <:ind>]
    get <attrib> [, …] into <:var>
    [[indicator] <:ind>] [, …];
exec sql [at <db>] [for <n>] collection append
    <:var> [[indicator] <:ind>] to <:coll>
    [[indicator] <:ind>];
exec sql [at <db>] collection trim <:x>
    from <:coll> [[indicator] <:ind>];
exec sql [at <db>] [for <n>] collection
    set <:coll> [[indicator] <:ind>]
    to <:var> [[indicator] <:ind>];
exec sql [at <db>] [for <n>] collection
    get <:coll> [[indicator] <:ind>]
    into <:var> [[indicator] <:ind>];
exec sql [at <db>] collection reset <:coll>
    [[indicator] <:ind>];

Objects

exec sql [at <db>] [for <n>] object
    create <:obj> [indicator] <:ind>
    [table <tab>] [returning ref into <:ret>];
exec sql [at <db>] [for <n>] object
    release <:obj>;
exec sql [at <db>] [for <n>] object
    update <:obj>;
exec sql [at <db>] [for <n>] object
    delete <:obj>;
exec sql [at <db>] [for <n>] object
    deref <:ref> into <:obj>
    [[indicator] <:ind>] [for update [nowait]];
exec sql [at <db>] [for <n>] object
    flush <:obj>;
exec sql [at <db>] object
    set [{ * | <attrib> [, …] } of]

<:obj> [[indicator] <:ind>] to <:var>
    [[indicator] <:ind>] [, …];
exec sql [at <db>] object
    get [{ * | <attrib> [, …] } from]
    <:obj> [[indicator] <:ind>] into <:var>
    [[indicator] <:ind>] [, …];
exec sql [for <n>] [object] free <:point>
    [[indicator] <:ind>];
exec sql [at <db>] [object] cache free all;

LOBs

exec sql [at <db>] lob append <:src> to <:dst>;
exec sql [at <db>] lob trim <:src> to <:x>;
exec sql [at <db>] lob assign <:src> to <:dst>;
exec sql [at <db>] lob copy <:x>
    from <:src> [at <:y>] to <:dst> [at <:z>];
exec sql [at <db>] lob
    erase <:x> from <:src> [at <:y>];
exec sql [at <db>] lob create temporary <:src>;
exec sql [at <db>] lob free temporary <:src>;
exec sql [at <db>] lob describe <:src> get
    { chunksize | directory | fileexists | filename
    | isopen | istemporary | length } [, …]
    into <:var> [[indicator] <:ind>] [, …];
exec sql [at <db>] lob enable buffering <:src>;
exec sql [at <db>] lob disable buffering <:src>;
exec sql [at <db>] lob flush buffer <:src> [free];
exec sql [at <db>] lob open <:src>
    [read only | read write];
exec sql [at <db>] lob close <:src>;
exec sql [at <db>] lob file close all;
exec sql [at <db>] lob file set <:file>
    directory = <:dir>, filename = <:name>;
exec sql [at <db>] lob load <:x> from file
    <:file> [at <:y>] into <:dst> [at <:z>];
exec sql [at <db>] lob read <:x> from <:src>
    [at <:y>] into <:buff> [with length <:z>];
exec sql [at <db>] lob write [append]
    [first | next | last | one] <:x>
    from <:buff> [with length <:y>]
    into <:dst> [at <:z>];

Context

exec sql context allocate <:cont>;
exec sql context use {<:cont> | default};
exec sql context object get <option> [, …]
    into <:var> [, …];
exec sql context object set <option> [, …]
    to <:var> [, …];
exec sql context free <:cont>;

Utilities

proc
    auto_connect={yes | no} char_
    map={varchar2 | charz | string | charf}
    close_on_commit={yes | no} code={ansi_c
    | kr_c | cpp} comp_charset={multi_byte
    | single_byte} config=<file> cpp_
    suffix=<ext> dbms={native | v7 | v8}
    def_sqlcode={yes | no} define=<name>
    duration={transaction | session}
    dynamic={oracle | ansi} errors={yes | no}
    errtype=<file> fips={sql89 | sql2 | yes
    | no} header=<ext> hold_cursor={yes
    | no} iname=<file> include=<path>
    intype=(<file, …>) lines={yes | no}
    lname=<lisfile> ltype={none | short | long}
    maxliteral=<1024> maxopencursors=<10>
    mode={ansi | iso | oracle} nls_char=<var>
    nls_local={yes | no} objects={yes |
    no} oname=<ofile> oraca={yes | no}
    pagelen=<80> parse={full | partial | none}
    prefetch=<1> release_cursor={yes | no}
    select_error={yes | no} sqlcheck={semantics
    | full | syntax} sys_include=<path>
    threads={yes | no} type_code={oracle |
    ansi} unsafe_null={yes | no} userid=<user>/
    <pwd>[@<serv>] utf16_charset={nchar_
    charset | db_charset} varchar={yes | no}
    version={recent | latest | any}

lnproc

SQLJ

```
#sql <mod> iterator <iter>
    [implements <intfc> [, …] ]
    [with ( [sensitivity = {sensitive
                | asensitive | insensitive}]
        [holdability = {true | false}]
        [returnability = {true | false}]
        [updatecolumns = '<col> [, …]
        [<var> = <val>] [, …] )]
    (<type> [<col>] [, …]);
```
named or positional iterator
```
#sql <mod> context <cont>
    [implements <intfc> [, …]]
    [with ( … <var>=<val> [, …] )];
#sql [ [<conn_cont_inst>, <exec_cont_inst>] ]
    [<var / iter> =] { <SQL stat> };
```
>> Curly braces are part of syntax! <<
```
#sql { select /*+ <HINT> */ <expr> [, …]
    into <:[out] var> [, …]
    from <tab> [where <expr> …] };
#sql <iter> = { select <expr> [, …]
    from <tab> [where <expr>…] };
#sql { fetch <:iter> into <:var> [, …] };
```
<iter>.next(), <iter>.endFetch(), <iter>.close()
```
#sql { insert into… };
#sql { update… };
#sql { delete from… };
#sql { commit };
#sql { rollback };
#sql { set transaction <mode>
    [, isolation level <level>] };
#sql { call <proc> (<par> [, …] ) };
#sql <var / iter> =
    { values ( <func> (<par> [, …] ) ) };
#sql { set <:var> = <expr> };
#sql <iter> = { cast <:result_set> };
#sql { [declare <var> <type>;]
    begin <stat>; […] end; };
```

Utilities

sqlj
```
    –d[ir]=<dir> –encoding=<enc> –url=<url>
    –status –compile=false –user=<user>/
    <pwd>@jdbc:oracle:thin@<host>:<port>:
    <sid> –linemap –profile=false –ser2class
    –P–<opt> –C–<opt> –P–help –C–help
    –J–<opt> –version –help–alias –help–log
    –<key>=<value>
    { <in>.sqlj [<out>.java] …
    | <in>.ser [<out>.jar] … }
```
loadjava
```
    –d[efiner] –e[ncoding] <latin1> –f[orce]
    –g[rant] <user / role>, … –h[elp]
    –noverify –order –r[esolve] –a[ndresolve]
    –s[ynonym] –oracleresolver
    –R[esolver] "((<name> <schema>) …)"
    –o[ci8] –t[hin] –v[erbose] <true>
    –S[chema] <schema>
    –u[ser] <user>/<pwd>@<netserv>
     <classes> <jars> <resources> <properties>
```
dropjava
```
    –encoding <latin1> –h[elp]–s[ynonym]
    –{o[ci8] | t[hin]} –v[erbose] –S[chema]
    <schema>  –user user/<pwd>@<netserv>
    <classes> <jars> <resources> <properties>
```
publish
```
    –republish –h[elp] –version –describe
    –g[rant] <user / role>, … –role <role>
    –user <user> –password <pwd> –service
    <url> –schema <schema>  –{ssl | iiop}
    <name> <class> [<helper>]
```
remove
```
    –r[ecurse] –h[elp] –version –d[escribe]
    –role <role> –user <user> –password
    <pwd> –service <url> –{ssl | iiop} <name>
```
sess_sh
```
    –h[elp] –version –d[escribe] –role <role>
```

```
    –user <user> –password <pwd> –service
    <url> –{ssl | iiop}
```
deployejb
```
    –generated <clientjar> –descriptor <file>
    –verbose –republish  –beanonly –addclass-
    path <path> –resolver <res> –h[elp] –keep
    –version –describe –p[roperties] <file>
    –user <user>  –password <pwd> –role
    <role> –service <url> –{ssl | iiop}
    –credsfile <file> –useservicename –temp
    <dir> <EJBjarfile>
```
ejbdescriptor
```
    –{parse | dump} <infile> <outfile>
```
java2rmi_iiop
```
    –no_bind –no_comments –no_examples
    –no_tie –wide –root_dir <dir> –verbose
    –version –W <n>
```
java2idl
modifyprops
```
    –{o[ci8] | t[hin]}
    –u[ser] <user>/<pwd>@<netserv>
    {<key> <val> | <key> –delete}
```

Label Security

Views & Tables

dba_sa_audit_options, dba_sa_compartments, dba_sa_data_labels, dba_sa_groups, dba_sa_group_hierachy, dba_sa_labels, dba_sa_levels, dba_sa_policies, dba_sa_prog_privs, dba_sa_schema_policies, dba_sa_table_policies, dba_sa_users, dba_sa_user_compartments, dba_sa_user_groups, dba_sa_user_labels, dba_sa_user_levels, dba_sa_user_privs

Packages

SA_SESSION

privs, {min | max}_level, comp_{read | write}, group_{read | write}, label, row_label, sa_user_name, {save | restore}_defaults_labels, set_label, set_row_label, set_access_profile, set_user_privs

SA_SYSDBA

{create | drop | enable | diable}_policy

SA_COMPONENTS

{create | alter | drop}_level, {create | alter |drop}_compartment, {create | alter | drop}_group, alter_group_parent

SA_LABEL_ADMIN

{create | alter | drop}_label

SA_POLICY_ADMIN

{apply | remove | enable | disable}_table_policy, {apply | alter | remove | enable | disable}_schema_policy

SA_USER_ADMIN

set_levels, {set | add | alter | drop}_compartments, {set | add | alter | drop}_groups, drop_all_{groups | compartments}, set_user_labels, set_{default | row}_label, set_prog_privs

SA_AUDIT_ADMIN

audit, noaudit, {audit | noaudit}_label, audit_label_enabled, {create | drop}_view

SA_UTL

{numeric | numeric_row | data}_label, set_{label | row_label}

Performance

Performance Gains

- minimal for instance tuning
- large for application tuning

Oracle Performance Improvement Method

1. user feedback: scope and goals?
2. get full set of OS, database, and application statistics for good and bad performance
3. sanity-check OS (over-used resources or hardware errors?)
   - CPU: user & kernel mode
   - disk: response times & queues
   - memory: paging
   - network: latency
4. check for top ten performance issues
5. build/refine conceptual model of system
   - inadequate single-user response time? points to application problems (investigate SQL and application internal statistics)
   - full CPU utilization? kernel mode (network or memory) user mode (non-database, database: top SQL)
   - serialization? investigate wait events
6. apply series of remedy actions ordered and stepwise
7. validate impact on statistics and user perception
8. repeat steps 5 to 7 until goals are met or become impossible

Emergency Performance Method

1. user feedback: throughput or response time problem? changes in environment?
2. hardware utilization (CPU, disk, memory, network)
3. constrained CPU or wait events?
4. emergency action to stabilize system
5. get reference data and perform detailed analysis

Top Ten Performance Issues

- connection management
- cursor sharing (bind variables, avoid dynamic SQL)
- I/O design (controller, bandwidth)
- redo log sizes and groups
- data block serialization (free lists, free list groups, block size, transaction slots, rollback segments)
- long full table scans (indexes, statistics)
- disk sorts
- recursive SQL (space management)
- schema errors and optimizer problems
- nonstandard initialization parameters (undocumented features)