Oracle 8i – Guide pratique de référence des commandes

Janvier 2005

Documentation technique # 26

sqlpac

INSTANCE

v$fixed_table, v$fixed_view_definition, v$indexed_fixed_column, v$instance, v$sga v$sgastat,, v$session, v$process, v$bgprocess, v$version, v$product_component_version, v$license, v$option, v$access, v$timer, v$parameter, v$parameter2, v$system_parameter, v$system_parameter2, v$obsolete_parameter, v$sql, v$sqlarea, v$sqltext, v$sqltext_with_newlines, v$sql_cursor, vsql_bind_data, vsql_bind_metadata, vsql_shared_memory, vlibrarycache, v$rowcache, v$rowcache_parent, v$object_dependency, v$rowcache_subordinate, v$open_cursor, v$db_object_cache, v$shared_pool_reserved, v$bh, v$cache, v$subcache, v$buffer_pool, v$buffer_pool_statistics, v$filestat, 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$mts, v$circuit, v$shared_server, v$dispatcher, v$dispatcher_rate, v$reqdist, v$queue, v$lock, v$enqueue_lock, v$locked_object, v$global_blocked_locks, v$session_connect_info, v$session_longops, v$system_cursor_cache, v$session_cursor_cache, v$session_object_cache, v$sess_io, vbsp, vpx_session, v$px_sesstat, v$px_process, v$px_process_sysstat, v$pq_sesstat, vpq_slave, vpq_sysstat, vpq_tqstat, vexecution, v$mls_parameters, deptree, session_context

'instance_name', 'service_names', 'db_block_size', 'db_block_buffers', 'buffer_pool_keep', 'buffer_pool_recycle', 'db_block_lru_latches', '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', 'mts_service', 'mts_dispatchers', 'mts_max_dispatchers', 'mts_servers', 'mts_max_servers', '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', 'use_indirect_data_buffers'

[*obsolete*: v$recent_bucket, v$current_bucket, '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']

init<sid>.ora

Background: SMON, PMON, DBW<*n*>, CKPT, LGWR, ARC<*n*>, SNP<*n*>, RECO, D<*nnn*>, S<*nnn*>, P<*nnn*>, RFS<*n*>, LCK<*n*>, QMN<*n*>, EMN<*n*>

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.

Packages DBMS_SYSTEM (set_sql_trace_in_session), DBMS_SESSION (set_sql_trace), DBMS_SHARED_POOL (keep, unkeep, sizes), DBMS_APPLICATION_INFO (set_module, set_action, set_client_info, read_module, read_client_info)

dbmspool.sql, prvtpool.plb, utlbstat.sql, utlestat.sql, catparr.sql, utldtree.sql

Tuning/Contention

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) sleeps / gets > 1% -> increase 'db_block_lru_latches' (max. CPU * 6 or BUFFERS / 50)

```
show parameter <string>
alter system set <param> = <value> [deferred];
```

mts_dispatchers = "{ (protocol = <*prot*>) | (description = (address =...)) | (address = (protocol = <*prot*>) (host = <*node*>) (port = <*port*>))} (connections = <*X*>) (dispatchers = <*1*>) (listener = <*list*>) ({pool | multiplex} = <*X*>) (ticks = <*15*>) (service = <*serv*>) (presentation = {ttc | oracle.aurora.server.{SGiopServer | GiopServer} }) ", mts_servers = <*X*>, resource_limit = {true | false}, global_names = {true | false}, scan_instances = <*X*>, cache_instances = <*X*>, license_max_sessions = <*X*>, license_sessions_warning = <*X*>, license_max_users = <*X*>, remote_dependencies_mode = {timestamp | signature}, resource_manager_plan

```
alter session set <param> = <value>;
```

optimizer_goal = {all_rows | first_rows | rule | choose}, sql_trace = {true | false}, global_names = {true | false}, skip_unusable_indexes = {true | false}, label = {'<*str*>' | dbhigh | dblow | oslabel}, mls_label_format = <*fm*>, flagger = {entry | immediate | full | off}, session_cached_cursors = <*X*>, close_cached_open_cursors = {true | false}, instance = <*X*>, parallel_instance_group = '<*gr*>', hash_area_size = <*X*>, hash_multiblock_io_count = <*X*>, remote_dependencies_mode = {timestamp | signature}, isolation_level = {serializable | read committed}, constraints = {immediate | deferred | default}, <*NLS_PARAMs*>, events '{10015 | 10046 | 10049 | 10210 | 10211 | 10212 | 10231 | 10235 | 10520} trace name context {forever, level <*X*> | off}', events 'immediate trace name {heapdump | controlf} level <*X*>'

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

```
alter system {enable | disable} restricted session;
alter system kill session '<SID>,<Serial#>';
alter system disconnect session '<SID>,<Serial#>' post_transaction;
```

```
orapwd file=<file> password=<pwd> entries=<X>
oradim –{new | edit | delete | startup | shutdown}
          –{sid <SID> | srvc <serv>} –newsid <SID>
          –usrpwd <pwd> –intpwd <pwd> –maxusers <X>
          –startmode {a | m} –shutmode {a | i | n}
          –{starttype | shuttype} {srvc | inst | srvc, inst}
          –pfile <par> –timeout <X>
tkprof <trc> <out> [explain=<user>/<pwd>@<netserv>] [table=<tab>]
        [print=<X>] [sys=no] [insert=<file>] [record=<file>] [aggregate=<N>]
        [sort=<opt>]
otrcfmt
oemctrl {start | stop} oms, oemapp console, vppcntl –start, vtm
```

DATABASE

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

'control_files', 'db_name', 'db_domain', 'db_files', 'compatible', 'read_only_open_delayed'

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

Tuning/Contention

phyrds, phywrts (v$filestat)

```
create database [<db>]
        [datafile '<file>' [,...] size <X> [reuse]
            [autoextend {on | off} [next <X> maxsize {<X> | unlimited} ] ] ]
        [logfile [group <X>] ('<log>' [,...] ) size <X> [reuse]
            [, [group <X>] ('<log>' [,...] ) size <X> [reuse] ] ... ]
        [controlfile reuse] [maxdatafiles <X>] [maxinstances <X>]
        [maxlogfiles <X>] [maxlogmembers <X>] [maxloghistory <X>]
        [character set <char>] [national character set <char>]
        [archivelog | noarchivelog] [exclusive];
```

```
alter database [<db>] rename global_name to <db>;
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];
alter database [<db>] backup controlfile to
        {'<file>' [reuse] | trace [resetlogs | noresetlogs] };
alter database [<db>] create standby controlfile as '<ctrl>' [reuse];
alter database [<db>] activate standby database;
dbassist
```

TABLESPACES, DATAFILES & SEGMENTS

v$tablespace, v$datafile, v$datafile_copy, v$datafile_header, v$dbfile, v$offline_range, v$tempfile, v$tempstat, 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, dba_tablespaces, dba_ts_quotas, dba_data_files, filext$, 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

'db_block_checking', 'db_block_checksum', 'recovery_parallelism', 'fast_start_io_target', 'fast_start_parallel_rollback', 'db_file_name_convert', 'log_checkpoint_interval', 'log_checkpoint_timeout', 'log_checkpoints_to_alert', 'db_writer_processes', 'db_file_simultaneous_waits', 'read_only_open_delayed', 'db_block_max_dirty_target'

[*obsolete*: 'db_file_simultaneous_writes', 'db_block_checkpoint_batch', 'parallel_transaction_recovery']

Packages DBMS_REPAIR (check_object, {skip | fix}_corrupt_blocks, dump_orphan_keys, rebuild_freelists, admin_tables)
DBMS_SPACE_ADMIN (tablespace_verify, tablespace_{rebuild|fix}_bitmaps, tablespace_migrate_{from | to}_bitmap, segment_{verify | corrupt | dump}, segment_drop_corrupt, segment_extent_map_dump)
DBMS_TTS (transport_set_check, downgrade)

```
create [temporary] tablespace <ts>
      {datafile | tempfile} '<file>' [size <X>] [reuse]
            [autoextend {off | on [next <X>] [maxsize {<X> | unlimited} ] } ]
      [,'<file>'... [autoextend...] ] [minimum extent <X>]
      [ default storage ( [initial <5xBS>] [next <5xBS >] [pctincrease <50>]
                          [minextents <1>] [maxextents {<X> | 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>] ] } ];
drop tablespace <ts> [including contents [cascade constraints] ];
```

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

```
alter system checkpoint [global | local];
alter system check datafiles [global | local];
alter database [<db>] {datafile | tempfile} '<file>' [,...] resize <X>;
alter database [<db>] {datafile | tempfile} '<file>' [,...] autoextend...;
alter database [<db>] datafile '<file>' [,...] {online | offline };
alter database [<db>] tempfile '<file>' [,...] {online | offline [drop] };
alter database [<db>] rename file '<file>' [,...] to '<new_file>' [,...];
alter database [<db>] create datafile '<file>' [,...] [as '<new>' [,...] ];
```

```
alter database [<db>] recover [automatic] [from '<log_path>']
        { database [until { cancel | change <scn>
                          | time '<YYYY-MM-DD:HH24:MI:SS>' } ]
                     [using backup controlfile]
        | [managed] standby database [timeout <X> | cancel [immediate] ]
        | [standby] tablespace '<ts>' [,...] [until [consistent with] controlfile]
        | [standby] datafile {'<file>' | <X>} [,...]
                                 [until [consistent with] controlfile]
        | logfile '<log>' | continue [default] | cancel }
        [noparallel | parallel [<X>] ];
set autorecovery {on | off}
set logsource <dir>
recover [automatic] [from '<log_path>']
        { database [until { cancel | change <scn>
                          | time '<YYYY-MM-DD:HH24:MI:SS>' } ]
                     [using backup controlfile]
        | [managed] standby database [timeout <X> | cancel [immediate] ]
        | [standby] tablespace '<ts>' [,...] [until [consistent with] controlfile]
        | [standby] datafile {'<file>' | <X>} [,...]
                                 [until [consistent with] controlfile]
        | logfile <log> | continue [default] | cancel}
        [ { noparallel | parallel (degree {<X> | default}
                              [instances <1> | default] ) } ]
dbv     file=<file> start=<X> end=<X> logfile=<log> blocksize=<2048>
        feedback=<0>
```

BLOCKS

v$type_size

Header: static(61B), row directory(2B*rec), transaction headers (23B*TX) [*Cluster:* table directory]

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}

Package 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

vlog, vlogfile, v$thread, v$loghist, v$log_history, v$database, v$archive, v$archive_dest, v$archived_log, v$archive_processes, v$logmnr_dictionary, v$logmnr_parameters, v$logmnr_logs, v$logmnr_contents, v$targetrba

'thread', 'log_buffer', 'log_archive_max_processes', 'log_archive_start', 'log_archive_dest', 'standby_archive_dest', 'log_archive_dest_[1-5]' = '{ location = <*path*> | service = <*serv*>} [optional | mandatory] [reopen [=<*300*>]]', 'log_archive_dest_state_[1-5]' = {enable | defer}, 'log_archive_duplex_dest', 'log_archive_min_succeed_dest', 'log_archive_format', 'log_file_name_convert', 'arch_io_slaves', 'utl_file_dir', '_allow_resetlogs_corruption' (undocumented & unsupported)

[*obsolete*: 'log_archive_buffers', 'log_archive_buffer_size', 'log_block_checksum', 'log_simultaneous_copies', 'log_small_entry_max_size', 'lgwr_io_slaves']

Packages DBMS_LOGMNR_D (build), DBMS_LOGMNR (add_logfile, start_logmnr, end_logmnr)

dbmslogmnrd.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)

```
archive log { list | stop | {start | next | all | <X>} } [to <dest>]
alter database [<db>] {archivelog | noarchivelog};
alter system archive log [thread <X>]
        { start [to '<log_path>'] | stop | current | next | all
        | sequence <X> | group <X> | change <X> | logfile '<file>' };
alter system switch logfile;
```

```
alter database [<db>] add logfile
                            [thread <X>] [group <X>] ('<log>',...) size <X>;
alter database [<db>] {enable [public] | disable} thread <X>;
alter database [<db>] add logfile member '<log>' [reuse] to group <X>;
alter database [<db>] rename file '<log>' [,...] to '<new_log>' [,...];
alter database [<db>] drop logfile group <X>;
alter database [<db>] drop logfile member '<log>';
alter database [<db>] clear [unarchived] logfile { group <X> | '<log>' }
                                  [unrecoverable datafile];
```

TABLES & CONSTRAINTS & TRIGGERS

dba_tables, dba_all_tables, dba_object_tables, 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_tabs, 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, hist_head$

'_system_trig_enabled'

[*obsolete*: *dba_histograms,* 'cache_size_threshhold']

Packages DBMS_UTILITY (analyze_database, analyze_schema, analyze_part_object)
DBMS_SPACE (unused_space, free_blocks) [dbmsutil.sql, utlvalid.sql]
DBMS_STATS (gather_{database | schema | table | index}_stats, {export | import}_schema_stats)

utlexcpt.sql, utlexcpt1.sql, dbmsstdx.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 (create as, delete from, insert into select, commit, drop)
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_segments -> recreate object)

full & partial partition-wise joins

```
create [global temporary] table <tab>                    (max. 1000 col)
    [of <object_type>]
        [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...]
          | unique [using index...]
          | check (<expr>)
          | references <tab> [ (<col>) ] [on delete {cascade | set null} ] }
        [ [not] deferrable [initially {immediate | deferred} ] ]
        [ {disable | enable} [validate | novalidate] [exceptions into <tab>] ]
    [,<col>... [constraint <col_constr>]...] [,...]
    [, constraint <tab_constr>...]
    [ref (<col>) with rowid] [scope for (<col>) is <tab>] )
    [tablespace <ts>] [organization {heap | index} ] [storage (...) ]
    [pctfree <10>] [pctused <40>] [initrans <1>] [maxtrans <X>]
    [logging | nologging]          [recoverable | unrecoverable] <- obsolete
    [cache | nocache] [monitoring | nomonitoring]
    [noparallel | parallel [<X>] ]
    [ partition by range (<col> [,...] )
    [ subpartition by hash (<col> [,...] )
        [subpartitions <X> [store in (<ts> [,...] ) ] ]
        ( partition <partX> values less than ( {<value> [,...] | maxvalue} )
                          [storage (...) ] [tablespace <ts>]
              [ ( subpartition <subpartX> [tablespace <ts>]
               [, subpartition...] ) ]
          [, partition... [ ( subpartition...) ] ] ) ]
    [ partition by hash (<col> [,...] )
        { ( partition <partX> [tablespace <ts>] [, partition...] )
```

```
          | partitions <X> store in (<ts> [,...] ) } ]
     [ {disable | enable} row movement]
     [ lob (<col>) store as
          ( [tablespace <ts>] [storage (...) ]
            [ {disable | enable} storage in row]
            [pctversion <10>] [chunk <X>]
            [cache | nocache {logging | nologging} ]
            [ index <ind> ([tablespace <ts>] [storage (...) ] ) ] ) ]   << deprecated
     [ varray <varr> store as lob [<lobseg>] ([tablespace <ts>] ) ]
     [nested table <col> store as <tab> [ ( (<prop>) [storage (...) ] ) ]
          [return as {locator | value} ]
     [on commit {delete | preserve} rows];
create table <tab> [logging | nologging] ... as select...;
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] [noparallel | parallel [<X>] ];
truncate table <tab> [ [preserve | purge] snapshot log]
                         [ {drop | reuse} storage];
alter table <tab> [storage (...) ] [noparallel | parallel [<X>] ] ...
       [ {nominimize | minimize} records_per_block];
alter table <tab> { allocate extent
                         ( [size <X>] [datafile '<file>'] [instance <X>] );
                    | deallocate unused [keep <X>] };
lock table <tab> in {share [row exclusive] | exclusive} mode [nowait];
alter table <tab> {enable | disable} table lock;
comment on {table <tab> | column <tab>.<col>} is '<str>';
```

```
alter table <tab> add partition <range_part>
       values less than (<value> [,...] ) [tablespace <ts>];
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 <part>
       [storage (...) ] [allocate extent...] [logging | nologging] ...
       [ [rebuild] unusable local indexes]
       [ add subpartition [<subpart> [tablespace <ts>] ]
       | coalesce subpartition];
alter table <tab> modify subpartition <subpart>
       [storage (...) ] [allocate extent...] [logging | nologging] ...
       [ [rebuild] unusable local indexes];
alter table <tab> modify default attributes
       [for partition <comp_part>] [storage (...) ] ...;
alter table <tab> move {partition | subpartition} <part>
       tablespace <ts> [parallel [<X>] ] [logging | nologging];
alter table <tab> split partition <part1> at (<X>)
       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];
```

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

```
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> }
        |   {   {create | alter | drop} [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>;
```

```
analyze table <tab> [partition(<X>) ]
                  { compute statistics
                  | estimate statistics [sample <1064> {rows | percent} ] }
                  [for table] [for all [local] indexes]
                  [for all [indexed] columns [size <75>] ]
                  [for columns <col> [size <75>] ];
analyze table <tab> delete statistics;
analyze table <tab> list chained rows [into <chained_rows>];
analyze table <tab> validate
                  { structure [into <invalid_rows>] [cascade]
                  | 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> [,...]
        | indexes <ind> [,...] | indextypes <indtype> [,...] } [force];
```

VIEWS & SYNONYMS & SEQUENCES

dba_views, dba_synonyms, dba_sequences

```
create [or replace] [force | no force] view <view> [ (<alias> [,...] ) ]
        [of <type> with object oid [default | (<attr>,...) ] ]
        as <query> [with { read only | check option [constraint <constr>] } ];
alter view <view> compile;
drop view <view>;
create [public] synonym <syn> for <obj>;
drop [public] synonym <syn>;
create sequence <seq> [start with <1>] [increment by <1>]
        [maxvalue <10^27> | nomaxvalue] [minvalue <1> | nominvalue]
        [cycle | nocycle] [nocache | cache <20>] [order | noorder];
alter sequence <seq> ...;
drop sequence <seq>;
```

CLUSTERS

dba_clusters, dba_clu_columns, all_tab_columns, dba_cluster_hash_expressions

```
create cluster <clus> (<col> <type> [,...] )
        [index | [single table] hashkeys <X> [hash is <expr>] ]
        [size <1xBS>] [tablespace <ts>] [storage (...) ]
        [pctfree <10>] [pctused <40>] [initrans <X>] [maxtrans <255>];
create index <ind> on cluster <clus>
        [storage (...) ] [tablespace <ts>] [pctfree <X>]
        [initrans <X>] [maxtrans <X>];
create table <tab>
        (<col> <type>... [constraint <constr>...] )
        cluster <clus> (<col> [,...] );
```

```
alter cluster <clus>...;
truncate cluster <clus> [ {drop | reuse} storage];
drop cluster <clus> [including tables [cascade constraints] ];
analyze cluster <clus> ...;
```

INDEX-ORGANIZED TABLES

all_tables (iot_type, iot_name), all_indexes

```
create table <iot> (<col>... primary key...)
        organization index
        [tablespace <ts>] [pctfree <X>] [initrans <X>] [maxtrans <X>]
        [storage (...) ] [pctthreshold <50> [including <col>] ]
        [compress [<X>] | 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 [<X>] | nocompress]
        [tablespace <ts>] [overflow...] ... [noparallel | parallel [<X>] ];
alter table <iot> modify default attributes [for partition <part>]
        [storage (...) ] [pctthreshold <50> [including <col>] ]
        [compress [<X>] | nocompress] [overflow tablespace <ts>...];
analyze table <iot> compute statistics;
```

INDEXES

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

'create_bitmap_area_size', 'bitmap_merge_area_size'

Package *DBMS_PCLXUTIL (build_part_index)*

Tuning

(index_stats) del_lf_rows_len / lf_rows_len > 20% -> rebuild index

```
create [unique | bitmap] index <ind>
      on <tab> ( [<expr>] <col> [asc | desc] [,...] )
      [tablespace {<ts> | default } ] [storage (...) ]
      [pctfree <10>] [initrans <X>] [maxtrans <255>]
      [logging | nologging] [nosort] [reverse] [online]
      [noparallel | parallel [<X>] ] [nocompress | compress [<X>] ]
      [ local
         [ { ( partition [<partX>] [storage (...) ] [tablespace <ts>]     on range p. tab
                  [logging | nologging] [, partition...] )
           | { store in ( {<ts> [,...] | default} )                          on hash p. tab
             | ( partition [<partX>] [tablespace <ts>] [, partition...] ) }
           | store in ( {<ts> [,...] | default} )                            on comp. p. tab
                [ ( partition [<partX>] [storage (...) ] [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_ind>;
```

```
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 <X>] ...;
alter index <ind> rebuild {partition | subpartition} <part>
        [tablespace <ts>] [parallel [<X>] ];
alter index <ind> split partition <part1> at values less than (<X>)
        into ( partition <part2>, partition <part3> [,...] );
```

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

ROLLBACK SEGMENTS

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

'rollback_segments', 'transactions', 'transactions_per_rollback_segment'

Package 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% or
               -> add RBS
RBS Segment: "%undo%" (v$waitstat) / "consistent gets" (v$sysstat) (count/value) > 1%
               -> add RBS
```

```
create [public] rollback segment <rbs> [tablespace <ts>]
        [storage ( [initial <5xBS>] [next <5xBS>] [optimal <null>]
                       [minextents <1>] [maxextents {<X> | unlimited} ] ) ];
drop rollback segment <rbs>;
```

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

TEMPORARY SEGMENTS

v$sort_segment, v$sort_usage, dba_segments

'sort_area_size', 'sort_area_retained_size', 'sort_multiblock_read_count'

[*obsolete*: 'sort_direct_writes', 'sort_write_buffers', 'sort_write_buffer_size']

Tuning

"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

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, 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

'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)

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

```
create role <role> [ [not] identified {by <pwd> | 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 };
```

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

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

```
        [logical_reads_per_session <X>] [private_sga <X>];
```

AUDITING

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, dba_audit_object, dba_audit_exists, audit_actions, sys.aud$

'audit_trail', 'transaction_auditing'

cataudit.sql, catnoaud.sql

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

NLS

v$nls_parameters, v$nls_valid_values, nls_database_parameters, nls_instance_parameters, nls_session_parameters, props$

Server: *init<SID>.ora*

```
NLS_LANGUAGE
    è   NLS_DATE_LANGUAGE, NLS_SORT
NLS_TERRITORY
    è   NLS_DATE_FORMAT
    è   NLS_CURRENCY (fm L), NLS_ISO_CURRENCY (fm C),
        NLS_DUAL_CURRENCY, NLS_UNION_CURRENCY
    è   NLS_MONETARY_CHARACTERS
    è   NLS_NUMERIC_CHARACTERS (fm DG)
    è   NLS_LIST_SEPARATOR
    è   NLS_CALENDAR
    è   NLS_CREDIT, NLS_DEBIT
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>]
lxegen
```

Client: *environment variables*

```
NLS_LANG, NLS_NCHAR
    è   NLS_DATE_LANGUAGE, NLS_SORT
    è   NLS_DATE_FORMAT
    è   NLS_CURRENCY, NLS_ISO_CURRENCY, NLS_DUAL_CURRENCY
    è   NLS_MONETARY_CHARACTERS
    è   NLS_NUMERIC_CHARACTERS
    è   NLS_CREDIT, NLS_DEBIT
    è   NLS_COMP
```

Session:

alter **session** set NLS_LANGUAGE=<*lang*> NLS_TERRITORY=<*territ*>;

Package DBMS_SESSION.SET_NLS(<*name*>,<*value*>)

SQL-Functions:

```
è   to_char (NLS_DATE_LANGUAGE,  NLS_NUMERIC_CHARACTERS,
             NLS_CURRENCY, NLS_ISO_CURRENCY, NLS_CALENDAR)
è   to_date (NLS_DATE_LANGUAGE, NLS_CALENDAR)
è   to_number (NLS_NUMERIC_CHARACTERS, NLS_CURRENCY,
               NLS_ISO_CURRENCY)
è   nls_upper (NLS_SORT)
è   nls_lower (NLS_SORT)
è   nls_initcap (NLS_SORT)
è   nlssort (NLS_SORT)
```

EXPORT & IMPORT & LOADS & MIGRATION

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

catexp.sql, catexp7.sql, migrate.bsq

```
exp     userid=<user>/<pwd> parfile=<par> file=<expdat.dmp>
        filesize=<X> volsize=<X> log=<log> buffer=<X> silent=<N>
        recordlength=<X> direct=<N> rows=<Y> indexes=<Y> grants=<Y>
        constraints=<Y> triggers=<Y> feedback=<0> inctype={complete |
        cumulative | incremental} statistics={estimate | compute | none}
        record=<Y> compress=<Y> consistent=<N> help=<N> { full=<N> |
        owner=<schema> | tables=(<tab>[:<part>] [,...] [query=<expr>] ) }
        transport_tablespace=<N> tablespaces=<ts> [,...]
        point_in_time_recover=<N> recovery_tablespaces=<ts> [,...]
imp     userid=<user>/<pwd> parfile=<par> file=<expdat.dmp>
        filesize=<X> volsize=<X> log=<log> buffer=<X> recordlength=<X>
        rows=<Y> grants=<Y> indexes=<Y> constraints=<Y> commit=<N>
        ignore=<N> inctype={system | restore} feedback=<0> show=<N>
        analyze=<Y> recalculate_statistics=<N> help=<N> destroy=<N>
        skip_unusable_indexes=<N> indexfile=<file> toid_novalidate=
        (<type> [,...] ) { full=<N> | tables=(<tab>[:<part>] [,...] ) }
        fromuser=<schema> [,...] touser=<schema> [,...]
        transport_tablespace=<N> datafiles='(<file> [,...] )'
        tablespaces=<ts> [,...] tts_owners=<owner> [,...]
        point_in_time_recover=<false>
        [Order: table creation – index defs – table data – B-tree index data
                    – triggers, constraints, bitmap indexes]
sqlldr userid=<user>/<pwd> data=<data> control=<ctrl> parfile=<par>
        log=<log>  bad=<bad> discard=<discard> discardmax=<X>
        skip=<X> load=<X> errors=<X> rows=<X> bindsize=<65536>
        readsize=<65536> silent=<Y> direct=<N> parallel=<N> file=<file>
        skip_unusable_indexes=<N> skip_index_maintenance=<N>
        commit_discontinued=<N>
mig     dbname=<db> new_dbname=<new> pfile=<initfile> spool=<logfile>
        check_only=<false> no_space_check=<false> multiplier=<15>
        nls_nchar=<char >
```

RECOVERY MANAGER

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$backup, v$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

'backup_tape_io_slaves', 'db_file_direct_io_count', 'disk_asynch_io', 'tape_asynch_io', 'control_file_record_keep_time'

[*obsolete*: 'arch_io_slaves', 'backup_disk_io_slaves', 'large_pool_min_alloc']

Packages DBMS_BACKUP_RESTORE (dbmssbkrs.sql, prvtbkrs.plb)
DBMS_RCVCAT,DBMS_RCVMAN (dbmsrman.sql, prvtrmns.plb)

catrman.sql, prgrmanc.sql

```
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>';
set snapshot controlfile name to '<new>';
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> ' '] ";
create catalog [tablespace <ts>];
upgrade catalog [tablespace '<ts>'];
drop catalog;
register database;
list incarnation of database;
reset database [to incarnation <id>];
resync catalog [from controlfilecopy ['<ctrl>'] ];
catalog { archivelog | datafilecopy | controlfilecopy} '<file>' [,...]
          [tag [=] '<tag>' | level [=] <X>];
change { archivelog | datafilecopy | backuppiece | backupset | proxy
          | controlfilecopy } {'<file>' | <X> | all | tag [=] '<tag>'}
          { delete | available | unavailable | uncatalog | validate
          | crosscheck };
{crosscheck | delete expired} backup [ of
        { { datafile | tablespace | database [skip tablespace] } '<name>'
        | controlfile | archivelog { all | like '<name>' | {from | until}
        { time [=] '<date>' | scn [=] <X> | logseq [=] <X> [thread = <X>] } } ]
        [ {tag = '<tag>' | completed { after | before} [=] '<date>'
                                        | between '<date>' and '<date>' } ];
allocate [auxiliary] channel <chann> [for {delete | maintenance} ]
        { type [=] {disk | '<dev>'} | name [=] '<name>' }
        [parms [=] "<par>"] [format [=] '<fm>']
        [connect [=] '<user>/<pwd>@<target_ops_inst>']
        [debug [=] <X>] [trace [=] <X>];
set limit channel <chann> [read rate [=] <X>] [kbytes [=] <X>]
                              [maxopenfiles [=] <X>];
release channel [<chann>];
report { { need backup { {incremental | days} | redundancy} [=] <X> }
          | unrecoverable }
                { datafile {'<file>' | <X>} [,...]
                  | tablespace '<ts>' [,...]
                  | database [skip tablespace '<ts>' [,...] ] }
        | obsolete { redundancy [=] <X> | orphan |
                     | until { time [=] '<date>' | scn [=] <X>
                               | logseq [=] <X> [thread [=] <X>] } }
        | schema [ at { time [=] '<date>' | scn [=] <X>
                        | logseq [=] <X> [thread [=] <X>] } ] }
        [device type {disk | '<dev>'} ];
list { {copy | backup} of
        { datafile {'<file>' | <X>} [,...]
        | tablespace '<ts>' [,...]
        | database [skip tablespace '<ts>' [,...] ]
        | controlfile
        | archivelog { all | like '<file>' | {from | until} { time [=] '<date>'
                             | scn [=] <X> | logseq [=] <X> [thread = <X>] } } }
        [tag [=] <tag>] [like '<string>'] [device type '<dev>']
        [recoverable [ until { time [=] '<date>' | scn [=] <X>
                               | logseq [=] <X> [thread [=] <X>] } ] ]
        [completed { {after | before} [=] '<date>'
                       | between '<date>' and '<date>' } ]
     | incarnation [of database ['<id>'] };
```

```
set maxcorrupt for datafile {'<file>' | <X>} to <X>;
copy   { datafile {'<file>' | <X>}
        | datafilecopy {'<file>' | tag [=] <tag>}
        | archivelog '<log>'
        | controlfilecopy {'<ctrl>' | tag [=] <tag>}
        | current controlfile }
        to '<dest>' [,...] [tag [=] '<tag>'] [level [=] <X>]
        [nochecksum] [check logical];
set duplex = {off | on | 1 | 2 | 3 | 4};
backup [full | icremental level [=] {0 | 1 | 2 | 3} ]
        [cumulative] [nochecksum] [check logical] [proxy [only] ]
     [(] { datafile {'<file>' | <X>} [,...]
        | datafilecopy {'<file>' | tag [=] <tag>} [,...]
        | tablespace '<ts>' [,...]
        | database
        | archivelog { all | like '<log>' | {from | until} { time [=] '<date>'
                          | scn [=] <X> | logseq [=] <X> [thread = <X>] } }
        | current controlfile
        | controlfilecopy '<ctrl>' }
        [include current controlfile] [delete input] [tag [=] <tag>]
        [format [=] '<fm>'] [filesperset [=] <X>] [channel <chann>]
        [skip {offline | readonly | inaccessible} ] [setsize [=] <X>]
        [diskratio [=] <X>] [pool [=] <X>] [parms [=] '<par>'] [ ) ];
validate backupset <X> [,...] [check logical];
set newname for datafile {'<file>' | <X>} to '<new>';
set archivelog destination to '<path>';
restore [(] { database [skip [forever] tablespace <ts> [,...] ]
             | tablespace '<ts>' [,...]
             | datafile {'<file>' | <X>} [,...]
             | archivelog { all | like '<log>' | {from | until} { time [=]
                 '<date>' | scn [=] <X> | logseq [=] <X> [thread = <X>] } }
             | controlfile [to '<ctrl>'] } [ ) ]
          [channel <chann>] [from tag [=] '<tag>'] [parms '<par>']
          [from {backupset | datafilecopy} ] [validate]
          [check readonly] [check logical]
          [ until { time [=] '<date>' | scn [=] <X>
          | logseq [=] <X> [thread [=] <X>] } ];
replicate controlfile from '<ctrl>';
switch datafile { {'<file>' | <X>}
                          [to datafilecopy {'<file>' | tag [=] <tag>} ]
                  | all };
set until { time [=] '<date>' | scn [=] <X> | logseq [=] <X> [thread [=] <X>] };
recover { database [ until { time [=] '<date>' | scn [=] <X>
                                | logseq [=] <X> [thread [=] <X>] } ]
                     [skip [forever] tablespace <ts> [,...] ]
          | tablespace '<ts>' [,...]
          | datafile {'<file>' | <X>} [,...] }
          [delete archivelog] [check readonly] [check logical] [noredo];
set auxname for datafile {'<file>' | <X>} to {'<new>' | null };
duplicate target database to '<db>'
        [ logfile { '<log>' [size <X>] [reuse]
                  | group <X> ('<log>' [,...] ) [size <X>] [reuse] } ]
        [nofilenamecheck] [skip readonly];
```

Net8 Middleware

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

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 = <*X*>) (oracle_home = <*path*>) (sid_name = <*SID*>) (program = <*prog*>) (prespawn_max = <*X*>) (prespawn_list = (prespawn_desc = (protocol = <*X*>) (pool_size = <*X*>) (timeout = <*X*>)))))), service_list_<*LISTENER*> = <*X*>, passwords_<*LISTENER*> = <*X*>, connect_timeout_<*LISTENER*> = <*X*>, use_plug_and_play_<*LISTENER*> = <*X*>, save_config_on_stop_<*LISTENER*> = <*X*>, trace_{level | file | directory}_<*LISTENER*> = <*X*>, logging_<*LISTENER*> = <*X*>, log_{file | directory}_<*LISTENER*> = <*X*>, startup_wait_time_<*LISTENER*> = <*X*>, queuesize = <*X*>, ssl_client_authentication = <*X*>, ssl_version = undetermined]

>> Since release 8.1 sid_list_<LISTENER> only required with enterprise manager! <<

sqlnet.ora

[log_{file | directory}_{client | server} = <*X*>, sqlnet.expire_time = <*X*>, use_cman = <*X*>, use_dedicated_server = <*X*>, sqlnet.{encryption | crypto_checksum}_{client | server} = {accepted | rejected | requested | required}, sqlnet.{encryption | crypto_checksum}_types_{client | server} = <*X*>, sqlnet.crypto_seed = <*X*>, trace_unique_client = <*X*>, trace_{level | file | directory}_{client | server} = <*X*>, tnsping.trace_{level | directory} = <*X*>, daemon.trace_{level | directory | mask} = <*X*>, sqlnet.authentication_services = <*X*>, sqlnet.client_registration = <*X*>, bequeath_detach = <*X*>, disable_oob = <*X*>, names.directory_path = ({hostname | tnsnames | onames | cds | nds | nis}), names.default_domain = <*X*>, name.default_zone = <*X*>, names.preferred_servers = <*X*>, names.initial_retry_timeout = <*X*>, names.request_retries = <*X*>, names.max_open_connections = <*X*>, names.message_pool_start_size = <*X*>, names.dce.prefix = <*X*>, names.nis.meta_map = <*X*>, namesctl.internal_encrypt_password = <*X*>, namesctl.internal_use = <*X*>, namesctl.no_initial_server = <*X*>, namesctl.noconfirm = <*X*>, namesctl.server_password = <*X*>, namesctl.trace_{level | file | directory | unique} = <*X*>]

[*obsolete*: automatic_ipc]

tnsnames.ora

[<net_serv> = (description = (address_list = (address = (protocol = <*X*>) (port = <*X*>) (host = <*node*>))) (connect_data = (service_name = <*serv*>) (instance_name = <*sid*>) (handler_name = <*X*>) (sdu = <*X*>) (server = dedicated) (hs = ok) (rdb_database = <*rdbfile*>) (type_of_service = <*X*>) (global_name = <*rdb*>) (failover_mode = (type = {select | session | none}) (method = {basic | preconnect}) (backup = <*serv*>))) (source_route = {on | off}) (failover = {on | off}) (load_balance = {on | off}))]

[*obsolete*: (connect_data = (sid = <*X*>))

>> Exception! Use of OEM and OPS on WinNT. Create net service names '<SID>_startup'. <<

names.ora

[names.server_name = <*X*>, names.addresses = <*X*>, names.region_checkpoint_file = <*X*>, default_domain = <*X*>, forwarding_available = <*X*>, log_file_name = <*X*>, log_stats_interval = <*X*>, reset_stats_interval = <*X*>, cache_checkpoint_interval = <*X*>, requests_enabled = <*X*>, server = <*X*>, namesctl_trace_level = <*X*>, trace_file_name = <*X*>, trace_level = <*X*>, names.trace_{file | directory | unique} = <*X*>, names.log_{file | directory} = <*X*>, queuesize = <*X*>]

[*obsolete*: names.use_plug_and_play, names.{domain | topology}_checkpoint_file]

protocol.ora

[<*prot*>.{excluded | invited}_nodes = <*node*>, <*prot*>.validnode_checking = <*X*>, tcp.nodelay = <*X*>]

cman.ora

[cman = (address = (protocol = <tcp>) (host = <*node*>) (port = <1630>)), cman_admin = (address = (protocol = <tcp>) (host = <*X*>) (port = <1830>)), cman_profile = (maximum_relays = <*X*>, relay_statistics = <*X*>, log_level = <*X*>, tracing = <*X*>, trace_directory = <*path*>, show_tns_info = <*X*>, use_async_call = <*X*>, authentication_level = <*X*>), cman_rules = (rule_list = (rule = (src = <*src*>) (dst = <*dst*>) (srv = <*serv*>) (act = accept | reject)))]

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

```
lsnrctl { start | stop | status | reload | set | show | help | change_password |
        services | save_config | dbsnmp_start | dbsnmp_stop |
        dbsnmp_status } [<LISTENER>]
namesctl { start | stop | reload | restart | status | ping <ns> | reorder_ns |
        start_client_cache | delegate_domain | domain_hint | flush |
        flush_name | log_stats | reset_stats | help | password | register |
        unregister | timed_query | query | repeat | set | show | version }
cmctl { start | stop | status | version } [cman | cm | adm]
trcasst [-o{c|d}{u[q]|t} -e[0|1|2] -s -p ...] <file>
netasst, tnsping, trcroute, adapters, oerr <tns> <errno>
```

Distributed DB / Replication / Heterogenous Services / Advanced Queuing / Data Warehousing

v$dblink, v$db_pipes, v$queue, v$aq, v$hs_agent, v$hs_session, v$hs_parameter, dba_db_links, dba_2pc_pending, dba_2pc_neighbors, dba_repcatlog, dba_repgroup, dba_repgroup_privileges, dba_repcolumn, dba_repcolumn_group, dba_repgenobjects, dba_repgrouped_column, dba_repkey_columns, dba_repsites, dba_repobject, dba_reppriority, dba_reppriority_group, dba_repprop, dba_repddl, dba_repconflict, dba_represolution, dba_represolution_method, dba_represol_stats_control, dba_represolution_statistics, dba_repparameter_column, dba_repcat_refresh_templates, dba_repcat_template_objects, dba_repcat_template_parms, dba_repcat_template_sites, user_repcat_temp_output, dba_repcat_user_authorizations, dba_repcat_user_parm_values, dba_jobs, dba_jobs_running, deftran, dba_snapshots, snap$, 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, 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*

'global_names', 'open_links', 'open_links_per_instance', 'distributed_transactions', 'commit_point_strength', 'job_queue_processes', 'job_queue_interval', 'aq_tm_processes', 'dblink_encrypt_login', 'replication_dependency_tracking', 'query_rewrite_enabled', 'query_rewrite_integrity', 'utl_file_dir', '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'

[*obsolete*: defcall, 'distributed_lock_timeout', 'snapshot_refresh_keep_connections', 'snapshot_refresh_processes', 'snapshot_refresh_interval', 'distributed_recovery_connection_hold_time', 'job_queue_keep_connections']

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}), 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_SNAPSHOT (purge_log, {begin | end}_table_reorganization, {register | unregister}_snapshot, set_i_am_a_refresh, i_am_a_refresh), 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 (refresh, refresh_all_mviews, refresh_dependent), 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)

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

```
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>';
```

```
create {materialized view | snapshot} log on <tab>
        [tablespace <ts>] [storage (...) ]
        [pctfree <10>] [pctused <40>] [initrans <1>] [maxtrans <X>]
        [logging | nologging] [cache | nocache] [noparallel | parallel [<X>] ]
        [partition ...] [lob ...] [using index...]
        [with [primary key] [, rowid] [ (<col> [,...] ) ] ]
        [ {including | excluding} 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 <10>] [pctused <40>] [initrans <1>] [maxtrans <X>]
        [logging | nologging] [cache | nocache] [noparallel | parallel [<X>] ]
        [cluster <clust> (<col> [,...] ) ] [lob...] [partition...]
        [build {immediate | deferred} ]
        [on prebuilt table [ {with | without} reduced precision] ]
        [using index...]
        [ {refresh [fast | complete | force] [on commit | on demand]
                [start with '<date>'] [next '<date>']
                [with {primary key | rowid} ]
                [using [default] [master | local] rollback segment [<rbs>] ] ]
        | never refresh } ]
        [for update]
        [ {enable | disable} query rewrite]
        as <query>;
alter {materialized view | snapshot} <mview> ... [compile];
drop {materialized view | snapshot} <mview>;
create [force | noforce] 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> [restrict | cascade]
                | hierachy <hier> | attribute <lev> }
        | compile };
drop dimension <dim>;
```

Parallel Server

gv$<DYN_PERF_VIEW>, v$active_instances, v$resource, v$resource_limit, v$ping, v$class_ping, v$file_ping, v$temp_ping, v$false_ping, v$lock_activity, v$lock_element, v$locks_with_collisions, v$lock_activity, v$lock_class_ping, v$cache_lock, v$dlm_latch, v$latch_misses, v$dlm_locks, vdlm_misc, vdlm_ress, vdlm_all_locks, vdlm_convert_local, vdlm_covert_remote, vdlm_traffic_controller, file_lock, ext_to_obj

'parallel_server', 'parallel_server_instances', 'thread', 'cpu_count', 'instance_number', 'instance_groups', 'parallel_instance_group', 'service_names', 'dml_locks', 'gc_files_to_locks' = "*<f#>* [-*<f#>*] = *<X>* [!*<B>*] [r] [each] [: ...]", 'gc_releasable_locks', 'gc_rollback_locks', 'gc_defer_time', 'gc_latches', 'lm_locks', 'lm_ress', 'lm_procs', 'max_commit_propagation_delay', 'parallel_default_max_scans', 'lock_name_space', [*obsolete*: *init_com.ora*, 'gc_latches', 'gc_lck_procs', 'delayed_logging_block_cleanouts', 'freeze_db_for_fast_instance_recovery', 'ogms_home', 'ops_admin_group']

init<db_name>.ora, <db_name>.conf, IDLM,PCM, OPQ, OPSM, OPSD

Background: LMON, LMD0, LCK*<n>*, BSP*<n>*

vendor OSDs: CM, Start, IO, IPC (RegKeys: CMDLL, IODLL, IPCDLL, STARTDLL)

catparr.sql

Tuning/Contention

Global cache:	'global cache %' (v$sysstat) *cache fusion latency:* ~ 1-40 ms 'global cache cr block receive time' / 'global cache cr blocks received' *consistent-read requests:* 'global cache cr block received' + 'global cache cr blocks read from disk' *average get time:* ~ 20-30 ms 'global cache get time' / 'global cache gets' *average convert time:* ~ 10-20 ms 'global cache convert time' / 'global cache converts' *timeouts:* 'global cache cr timeouts', 'global cache convert timeouts' = 0
Global locks:	'global lock %' (v$sysstat) *average global lock gets:* ~ 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') *IDLM non-PCM resources:* v$librarycache, v$rowcache
IDLM:	*lock statistics:* v$dlm_convert_local, v$dlm_convert_remote *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' + 'remote instance undo block writes') / 'DBWR forced writes' (v$sysstat)
Locking:	'releasable freelist waits" (v$sysstat)
Lock convers.:	*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 content:	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 = *<X>*;')
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>*;'

```
opsctl [start | stop] –c<user>/<pwd> –n<db> [–i<sid> [,...] ]
        [–f] [–t] [–u] [–m] [–y | e] [–v] [–h]
setlinks /f:<file> /d
```

Fail Safe

```
fscmd { dumpcluster | movegroup | onlinegroup | offlinegroup
        | onlineresource | offlineresource | verifygroup | verifyallgroups }
        <resource> /cluster = <clust> [ /logfile = <log>] [ /node = <node>]
        [ /offline = { abort | immediate | transactional | normal } ]
        [ /domain = <domain>  /user = <user>  /pwd = <pwd>]
```

SQL *Plus

appi[nfo], array[size], auto[commit], autop[rint], autorecovery, autot[race] [trace][only] exp[lain] stat[istics], blo[ckterminator], cmds[ep], colsep, com[patibility], con[cat], copyc[ommit], copytypecheck, def[ine], describe {depth | indent | linenum}, echo, editf[ile], emb[edded], esc[ape], feed[back], flagger, flu[sh], hea[ding], heads[ep], instance, lin[esize], lobof[fset], logsource, long, longc[hunksize], newp[age], null, numf[ormat], num[width], pages[ize], pau[se], recsep, recsepchar, serverout[put], shift[inout], show[mode], sqlbl[anklines], sqlc[ase], sqlc[ontinue], sqln[umber], sqlpre[fix], sqlp[rompt], sqlt[erminator], suf[fix], tab, term[out], ti[me], timi[ng], trim[out], trims[pool], und[erline], ver[ify], wra[p] sql.pno, sql.lno, sql.release, sql.sqlcode, sql.user

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

```
/, 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
{ho[st] | ! | $} <command>
store [set] <file> [ cre[ate] | rep[lace] | app[end] ]
def[ine] <var> = <value>, undef[ine] <var>
pro[mpt] [<string>], pau[se] [<string>]
conn[ect] [internal] [as {sysdba | sysoper} ], disc[onnect]
passw[ord] [<user>]
recover
```

```
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} <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 | <X> | <var> | <:var>]
        [commit | rollback]
```

```
{tti[tle] | bti[tle] | reph[eader] | repf[ooter]}
        [ { [page] [ le[ft] | ce[nter] | r[ight] ] [col <X>] [tab <X>] [bold]
        [s[kip] <X>] [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] <X> | 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] }
```

```
copy [from <user>@<db>] [to <user>@<db>]
        {create | replace | insert | append} <tab> [ (<col>,...) ]
        using <query>;
```

Data Types SQL*Plus

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

DATA TYPES (PL/SQL & DB columns)

v$type_size, v$temporary_lobs, 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

Scalar:

character	char(*<1>*) {*pl*: 32.767B, *col:* 2.000B}
	(Subtype: character)
	varchar2(*<X>*) {*pl*: 32.767B *(preallocated < 2000B)*, *col:* 4.000B} (Subtypes: string, varchar)
	nchar(*<1>*) {*pl*: 32.767B, *col:* 2.000B}
	nvarchar2(*<X>*) {*pl*: 32.767B, *col:* 4.000B }

binary_integer	{-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	{-2.147.483.647 .. 2.147.483.647} *machine arithmetic*
number(*<p>*,*<s>*)	{38digits = 21B (20B Mantisse, 1B Exponent)} (Subtypes: dec, decimal, double precision, float, int, integer, numeric, real, smallint)
date	{7B=CentYearMonDayHourMinSec, -4.712 to 9.999}
raw	{*pl*: 32.767B, *col:* 2.000B}
rowid	{ext: 10B, restr. 6B} *col:* urowid [(*<4000B>*)] {*physical and logical rowids*}

large objects:	long {*pl*: 32.760B, *col:* 2^31-1B=2G},
	long raw {*pl*: 32.760B, *col:* 2^31-1B=2G}
	internal: BLOB, CLOB, NCLOB {2^32-1B=4G}
	external: BFILE{pointer}
	create [or replace] directory *<dir>* as '*<path>*';
	drop directory *<dir>*;

boolean	{true | false | null}

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

Relationship:	
ref	ref cursor, ref *<otype>* {*pointer*}

Record:	*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	*may 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: 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} (<X>) 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

Implicit Conversions

	bin_int	char	date	long	number	pls_int	raw	urowid	varchar2
bin_int		X		X	X	X			X
char	X		X	X	X	X	X	X	X
date		X		X					X
long		X					X		X
number	X	X		X		X			X
pls_int	X	X		X	X				X
raw		X		X					X
urowid		X							X
varchar2	X	X	X	X	X	X	X	X	

Explicit Conversions

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

	char, varchar2	number	date	raw	rowid, urowid	nchar, nvarchar2
char, varchar2	X	X	X	X	X	
number	X	X			X	
date	X		X			
raw	X			X		
rowid, urowid	X					
nchar, nvarchar2		X	X	X	X	X

SQL & PL/SQL & Embedded SQL & Java / SQLJ

v$reserved_words, dba_source, dba_errors, dba_dependencies, deptree, ideptree, dba_libraries, dba_outlines, dba_outline_hints, outln.ol$, outln.ol$hints, java$options, java$class$md5$table, create$java$lob$table

'optimizer_mode', 'db_file_multiblock_read_count', 'optimizer_percent_parallel', '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', 'hash_multiblock_io_count', 'star_transformation_enabled', 'always_anti_join', 'always_semi_join', 'row_locking', 'sql_trace', 'timed_statistics', 'create_stored_outlines', 'use_stored_outlines', 'utl_file_dir', 'plsql_v2_compatibility', 'remote_dependencies_mode',
[*obsolete*: 'fast_full_scan_enabled', 'push_join_predicate']

Packages DBMS_STANDARD, DBMS_TRACE, DBMS_LOCK, DBMS_DESCRIBE, 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_cursor, parse, bind_variable, define_column, execute, fetch_rows, column_value, variable_value, close_cursor, is_open, execute_and_fetch, last_error_position, last_row_{count | id}, last_sql_function_code), UTL_FILE, UTL_HTTP, OUTLN_PKG (drop_unused, {drop | update}_by_cat, drop_{collision | extras | unrefd_hints}[_expact]), deptree_fill, DBMS_TRANSACTION (begin_discrete_transaction)
[>> Discrete transactions do not generate undo information! <<]
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}), SQLJUTL (has_default)

utldtree.sql, initjvm.sql, utljavarm.sql, sqljutl.sql

Functions and Operators

avg, count, sum /min, max, grouping, variance, stddev, round, trunc, ceil, floor, abs, sign, mod, cos, sin, tan, cosh, sinh, tanh, acos, asin, atan, atan2, exp, ln, log, power, sqrt, nvl, greatest, least, dump, bfilename, empty_[b|c]lob, uid, user, userenv, sys_guid, sys_context, vsize, nls_charset_name, nls_charset_id, nls_charset_decl_len, convert, to_number, to_char, to_date, to_single_byte, to_multi_byte, to_lob, hextoraw, rawtohex, chartorowid, rowidtochar, decode, ascii, instr, instrb, length, lengthb, substr, substrb, upper, lower, trim, ltrim, rtrim, lpad, rpad, replace, translate, initcap, concat, chr, nls_upper, nls_lower, nlssort, nls_initcap, soundex, sysdate, next_day, last_day, add_months, months_between, new_time, ref, deref, make_ref, reftohex, value, multiset, cast, level, prior

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

Analytic Functions

Ranking:

```
{ rank() | dense_rank() | cume_dist() | percent_rank() | ntile(<X>) | 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 <X> | unbounded | interval '<X>' day} preceding]
                [ { [and] <X> | unbounded | interval '<X>' 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} ] )
```

DDL alter, create, drop, audit, noaudit, grant, revoke, update, truncate, rename, analyze, comment

DML select, insert, delete, lock table, explain plan

TxCtl commit, rollback, savepoint, set transaction

SessCtl alter session, set role

SysCtl alter system

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, 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>* [, {*<X>* | default} [{*<X>* | default}]]), NOPARALLEL (*<tab>* [...]), PQ_DISTRIBUTE (*<tab>* [,] *<out>*, *<in>*), APPEND, NOAPPEND, PARALLEL_INDEX (*<tab>* [*<ind>* [,...]] [, {*<X>* | default} [{*<X>* | 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_TRANSFORMATION, ORDERED_PREDICATES

Serial direct-load insert:

```
        insert /*+APPEND */ into <tab> <query>;
```

Parallel direct-load insert:

```
        alter session {enable | force} parallel dml;
        insert /*+PARALLEL(<tab>,<X>) */ into <tab> <query>;
```

```
select    { [aggr_func ( ]  [ distinct | unique | all ] { [<alias>.]<col> | * }  [ ) ]
                              [ {+ | - | * | /} <expr>] [as] ["<alias>"] [,...]
            | <seq>.{nextval | currval}
            | cursor (<subquery>) }
            [ [bulk collect] into <var> [,...] ]
        from { [<schema>.]<tab/view/snapshot>[@<dblink>]
                    [partition (<part>) ] [subpartition (<subpart>) ] [<alias>] [,...]
                    [sample [block] (<X>) ]
                | ( <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> | <X>} [asc | desc] [,...] ]
        [for update [of <tab>.<col>] [nowait] ];
```

```
insert into { <tab> [partition (<part>) ] | [the] <subquery1>}
        [ (<col> [,...] ) ] {values (<expr>,...) | <subquery2>} [ref into <item>];
update <tab> set <col> = {<val> | '<string>'} [,...];
delete [from] <tab> [partition (<part>) ] [alias] [where <expr>];
```

```
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>};
alter session {enable | disable} commit in procedure;
alter session {enable | disable | force} parallel {dml | ddl} [parallel <X>];
```

```
declare
        -- <comment>
        /* <comment> */
        pragma autonomous_transaction;
        pragma serially_reusable;
        pragma restrict_references (<name>,
                                        rnds, wnds, rnps, wnps, trust);
        pragma exception_init (<exc>, <err_no>);
        <var> [constant]
            {<type> | <tab>.<col>%TYPE | <var>%TYPE | <tab>%ROWTYPE}
            [ [not null] { := | default } <X> ];
        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> >> ]
        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>
            <stat> [returning <col> bulk collect into <collect>];
            SQL%bulk_rowcount(i)
        open <curs> [ (<par>,...) ]; <curs>%found, %isopen, %notfound, %rowcount
        fetch <curs> [bulk collect] into {<var> [,...] | <rec>};
        close <curs>;
        open <refcurs> for <query> [using <var> [,...] ];
        execute immediate '<dyn_sql>' [into {<var> [,...] | <rec>} ]
                    [using [in | out | in out] <arg> [,...] ];
        [ << <label> >> ]
        goto <label>;
        insert into {<tab> | table (<subquery>) }
                [ (<col> [,...] ) ] {values (<expr>,...) | <subquery>}
                [returning <expr> [,...] into <var> [,...] ];
        update {<tab> | table (<subquery>) }
                set <col> = <expr> [,...]
                [where {<expr> | current of <curs>} ]
                [returning <expr> [,...] into <var> [,...] ];
        delete from {<tab> | table (<subquery>) }
                [where {<expr> | current of <curs>} ]
                [returning <expr> [,...] into <var> [,...] ];
        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;
        <proc> ( [<form_par> => ] <act_par> [,...] );
        return [ [(] <expr> [)] ];
        raise <exc>;
        exception
            when {<exc> | others} [or <exc2> ...] then
                <stat>; [sqlcode; sqlerrm(<X>);]
                raise;
        >> Predefined Server Exceptions:
        no_data_found, too_many_rows, invalid_cursor, zero_divide, dup_val_on_index <<
end;
```

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

```
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];
```

```
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>] ];
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>";
```

```
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>;
```

```
exec oracle define <symbol>;
exec oracle {ifdef | ifndef} <symbol>;
exec oracle {else | endif};
exec oracle option (oraca=yes);
exec sql include {oraca | sqlca};                sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc
exec sql declare <db> database;
exec sql connect {<:user> identified by <:pw> | <:user_pw>}
        [ [at {<db> | <:host>} ] using <:db>];
exec sql declare <stat> statement;
exec sql declare <tab> table (<col> <type>
        [ default <expr> [null | not null] | not null [with default] ]... );
exec sql [for <:var>] allocate descriptor
        [global | local] {<:descr> | '<descr>'} [with max <X>];
exec sql deallocate descriptor [global | local] {<:descr> | '<descr>'};
exec sql describe [input | output] <stat> using [sql] descriptor
        [global | local] {<:descr> | '<descr>'};
exec sql [for <:var>] set descriptor [global | local] {<:descr> | '<descr>'}
        { count = <:var> | value <item> [ref] <item> = <:var> [,...] };
exec sql [for <:var>] get descriptor [global | local] {<:descr> | '<descr>'}
        { <:var> = count | value <item> <:var> = <item> [,...] };
exec sql prepare <stat> from {<:str> | '<str>'};
exec sql [for <:var>] execute <stat>
        [using { <:var> [ [indicator] <:ind>] [,...]
                | [sql] descriptor [global | local] {<:descr> | '<descr>'}
                  [into [sql] descriptor [global | local] {<:descr> | '<descr>'}] } ];
exec sql execute immediate {<:str> | '<str>'};
exec sql execute begin <stat>; end; end-exec;
exec sql declare <curs> cursor for <query>;
exec sql var <:var> is <type>;
exec sql [for <:var>] open <curs>
        [using { <:var> [ [indicator] <:ind>] [,...]
                | [sql] descriptor [global | local] {<:descr> | '<descr>'}
                  [into [sql] descriptor [global | local] {<:descr> | '<descr>'}] } ];
exec sql {allocate | close} <curs>;
exec sql [for <:var>] fetch <curs>
        into { <:var> [ [indicator] <:ind>] [,...]
            | [sql] descriptor [global | local] {<:descr> | '<descr>'} };
exec sql select <val> into <:var>... from <tab> where <expr>...;
exec sql insert into {<tab> | (<subquery1>)}
        [ (<col>,...) ]  {values (<expr>,...) | <subquery2>};
exec sql update <tab> set <col> = <expr> [where <expr>];
exec sql [for <:host>] delete [from] <tab> [alias]
        [where {<expr> | current of <curs>} ];
```

```
exec sql describe [bind variables for | select list for} <stat> into <descr>;
exec sql [at {<db> | <:host>} ] commit [work];
        [ { [comment '<str>'] [release] | force '<id>' [,<X>] } ];
exec sql savepoint <sp>;
exec sql rollback [work] [to [savepoint] <sp> [release] | public];
exec sql whenever {not found | sqlerror | sqlwarning}
        {continue | goto <label> | stop | do {<routine> | break | continue} };
```

```
#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; };
```

```
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] –oracleresolver
           –R[esolver] "( (<name> <schema>) ... )" –s[ynonym]
           –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 –addclasspath <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 <X>
java2idl
modifyprops     –{o[ci8] | t[hin]} –u[ser] <user>/<pwd>@<netserv>
                {<key> <val> | <key> –delete}
```