```
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$sql_cursor, v$sql_bind_data, v$sql_bind_metadata,
v$sql shared memory, v$librarycache, v$rowcache, v$rowcache parent,
v$object_dependency, v$rowcache_subordinate, v$open_cursor,
/$db_object_cache, v$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$dispatcher rate, v$regdist, 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$sess_io, v$bsp,
v$px_session, v$px_sesstat, v$px_process, v$px_process_sysstat,
v$pq_sesstat, v$pq_slave, v$pq_sysstat, v$pq_tqstat, v$execution,
v$mls_parameters, deptree, session_context
'instance name', 'service names', 'db block size', 'db block buffers', 'buffer pool keep'
'buffer pool recycle', 'db block Iru 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 sqa', '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' ]
initesid> 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
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$rowcache)
               -> increase 'shared_pool_size'
              "cache buffers Iru chain" (v$latch) sleeps / gets > 1%
              -> increase 'db_block_Iru_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
   = <\underline{1}>) (listener = <list>) ( {pool | multiplex} = <X>) (ticks = <\underline{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
       I 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>]
oemctrl (start | stop) oms, oemapp console, vppcntl -start, vtm
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$,
database compatible level
'control_files', 'db_name', 'db_domain', 'db_files', 'compatible', 'read_only_open_delayed' catalog.sql, catproc.sql, u0703040.sql, r0703040.sql, u080<X>0.sql, r08000<X>0.sql,
d080<X>0<X>.sql, utlrp.sql, utlip.sql, utlirp.sql, utlconst.sql, utlincmpt.sql, utldst.sql,
catlg803.sgl
Tuning/Contention
phyrds, phywrts (v$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 = \langle str \rangle \mid dbhigh = \langle str \rangle \mid dbmac \{on \mid 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;
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_Imt_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' [obsolate: 'db_file_simultaneous_writes', 'db_block_checkpoint_batch',
           'parallel_transaction_recovery' ]
Packages DBMS_REPAIR (check_object, {skip | fix}_corrupt_blocks, dump_orphan_keys,
           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]

'<file>'... [autoextend...] [minimum extent <X>]

[autoextend {off | on [next <X>] [maxsize {<X> | unlimited}] }]

```
[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 }
       [\underline{noparallel} \mid 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])}]
      file=<file> start=<X> end=<X> logfile=<log> blocksize=<2048>
BLOCKS
v$type_size
              static(61B), row directory(2B*rec), transaction headers
              (23B*TX) [Cluster: table directory]
ROWID
               hex string of variable length
Logical:
Extend(10B): DataObj#{32b} - RelFile#{10b} - Block#{22b} - Row#{16b}
Base64
               000000 - FFF - BBBBBB - RRR
               Block#{Xb} - Row#{Xb} - File#{Xb}
Restrict(6B):
          DBMS ROWID
          (rowid create, rowid object, rowid relative fno, rowid block number,
          rowid_row_number, rowid_to_absolute_fno, rowid_to_extended,
          rowid to restricted)
v$log, v$logfile, v$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}
```

```
hist head$
                                                                                          svstem tria enabled
                                                                                                     import) schema stats)
                                                                                          utlexcpt.sql, utlexcpt1.sql, dbmsstdx.sql
                                                                                          Tuning/Contention
                                                                                                     "table scans%" (v$sysstat)
                                                                                                     -> increase octfree
                                                                                                     -> alter pctfree/pctused, inittrans, or
                                                                                                     -> increase freelist/freelist groups
                                                                                          full & partial partition-wise joins
                                                                                          create [global temporary] table <tab>
                                                                                              [of <object_type>]
                                                                                                  [with rowid] [scope is <tab>]
                                                                                                   [constraint < col_constr>]
                                                                                                   [ { [not] <u>null</u>
                                                                                                     primary key [using index...]
                                                                                                      unique [using index...]
                                                                                                     check (<expr>)
                                                                                               [, constraint <tab_constr>...]
                                                                                               [logging | nologging]
                                                                                               [noparallel | parallel [<X>]]
                                                                                               [ partition by range (<col> [,...] )
                                                                                               [ subpartition by hash (<col> [,...])
log_archive_duplex_dest', 'log_archive_min_succeed_dest', 'log_archive_format',
[obsolete: 'log_archive_buffers', 'log_archive_buffer_size', 'log_block_checksum',
                                                                                                          [, subpartition...])]
Packages DBMS_LOGMNR_D (build), DBMS_LOGMNR (add_logfile, start_logmnr,
                                                                                              [ partition by hash (<col> [,...])
```

```
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,
[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 |
pctfree = UPD/AVG, pctused = 1 - pctfree - AVG/nBLK
           -> adjust 'db_file_multiblock_read_count'
          table_name, head_rowid (chained_rows <- utlchain.sql, utlchain1.sql) or
           "table fetch continued row" (v$sysstat)
           -> recreate table (create as, delete from, insert into select, commit, drop)
           "segment header" (v$waitstat), "buffer busy waits" (v$system_event)
             (v$session_wait -> dba_extents -> dba_segments -> recreate object)
                                                               (max. 1000 col)
         [object identifier is {primary key | system generated
              [oidindex <ind> ( [tablespace <ts>...] [storage (...) ] ) ] } ]
    ( <col> <type> [ {default | := } <value>]
           references <tab> [ (<col>) ] [on delete {cascade | set null} ] }
         [ [not] deferrable [initially {immediate | deferred} ] ]
         [ {disable | enable} [validate | novalidate] [exceptions into <tab>] ]
     [,<col>... [constraint <col_constr>]...] [,...]
     [ref (<col>) with rowid] [scope for (<col>) is <tab>])
     [tablespace <ts>] [organization {heap | index} ] [storage (...) ]
     [pctfree <\underline{10}>] [pctused <\underline{40}>] [initrans <\underline{1}>] [maxtrans <X>]
                                     [recoverable | unrecoverable] <- obsolete
     [cache | nocache] [monitoring | nomonitoring]
         [subpartitions <X> [store in (<ts> [,...])]]
          ( partition <partX> values less than ( {<value> [,...] | maxvalue} )
                         [storage (...) ] [tablespace <ts>]
                [ ( subpartition <subpartX> [tablespace <ts>]
            [, partition... [ ( subpartition...) ] ] ) ]
         { ( partition <partX> [tablespace <ts>] [, partition...] )
```

[default storage ([initial <5xBS>] [next <5xBS >] [pctincrease <50>] 25-Jan-2002

'log file name convert', 'arch io slaves', 'utl file dir'.

end_logmnr)

_allow_resetlogs_corruption' (undocumented & unsupported)

log_simultaneous_copies', 'log_small_entry_max_size', 'lgwr_io_slaves']

Oracle Server 8i Quick Reference Card

```
| 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 (...) ]
                  [\underline{logging} \mid nologging] [\underline{noparallel} \mid 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...]
```

25-Jan-2002

```
| {shutdown | startup | servererror | logon | logoff} [or...] }
          on {schema | database} }
       [referencing
           {old [as] <old> | new [as] <new> | parent [as] <parent>} [,...] ]
       [for each row] [when (<expr>)]
        i cali ... :)
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;
create [public] synonym <syn> for <obj>;
drop [public] synonym <syn>;
create sequence <seq> [start with <1>] [increment by <1>] [maxvalue <10^{-1}> | nomaxvalue] [minvalue <1> | nominvalue]
       [cycle | \underline{\text{nocycle}} [nocache | cache <\underline{20}>] [order | noorder];
alter sequence <seq> ...;
drop sequence <seq>;
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 <\underline{10}>] [pctused <\underline{40}>] [initrans <X>] [maxtrans <\underline{255}>];
create index <ind> on cluster <clus>
       [storage (...) ] [tablespace < t s >] [pctfree < X >]
       [initrans <X>] [maxtrans <X>];
       (<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...)
       [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,
 create_bitmap_area_size', 'bitmap_merge_area_size'
              DBMS_PCLXUTIL (build_part_index)
(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>] ]
       [ { ( 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} )
              [ ( 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 \langle ts \rangle] [parallel [\langle X \rangle]];
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>];
       [ {partition | subpartition} <part>] [tablespace <ts>] [storage (...) ]
       [pctfree <\underline{10}>] [initrans <X>] [maxtrans <\underline{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;
 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
 RBS Segment: "%undo%" (v$waitstat) / "consistent gets" (v$sysstat) (count/value) > 1%
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']
<u>Tuninq</u> "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
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 > ...;
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];
\label{eq:cobject} \text{revoke} \ \{ \ \textit{<priv>} \ [ \ (\textit{<col>} \ [,...] \ ) \ ] \ | \ \text{all} \ [ \text{privileges} ] \ \} \ \text{on} \ [ \text{directory} \ | \ \textit{<object>} \ ]
        from { <user> | <role> | public } [cascade constraints]
create profile crof> 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>]
```

Oracle Server 8i Quick Reference Card

```
[logical_reads_per_session <X>] [private_sga <X>];
                                                                                                                                                                - triggers, constraints, bitmap indexes]
                                                                                                                                         sqlldr userid=<user>/<pwd> data=<data> control=<ctrl> parfile=<par>
                                                                                                                                                     log=<log> bad=<bad> discard=<discard> discardmax=<X>
                                                                                                                                                     skip = \langle X \rangle load = \langle X \rangle errors = \langle X \rangle rows = \langle X \rangle bindsize = \langle 65536 \rangle
all_def_audit_opts, dba_stmt_audit_opts, stmt_audit_option_map,
                                                                                                                                                     readsize=<65536> silent=<Y> direct=<N> parallel=<N> file=<file>
dba_priv_audit_opts, dba_obj_audit_opts, user_tab_audit_opts,
                                                                                                                                                     skip_unusable_indexes=<<u>N</u>> skip_index_maintenance=<<u>N</u>>
dba_audit_trail, dba_audit_session, dba_audit_statement, dba_audit_object,
                                                                                                                                                     commit_discontinued=<N>
dba_audit_exists, audit_actions, sys.aud$
                                                                                                                                                     dbname=<db> new_dbname=<new> pfile=<initfile> spool=<logfile>
'audit trail', 'transaction auditing'
                                                                                                                                                     check_only=<false> no_space_check=<false> multiplier=<15>
cataudit.sql, catnoaud.sql
                                                                                                                                                     nls nchar=<char>
[no]audit {<stat> [,...] | <priv> [,...] } [by <user> [,...] ]
          [by {session | access}] [whenever [not] successful];
                                                                                                                                          RECOVERY MANAGER
           (shortcuts: user, table, procedure, resource, connect, dba,...)
[no]audit <stat> [,...] on {<object> | default}
                                                                                                                                          rc_database, rc_database_incarnation, rc_backup_set, rc_backup_piece,
                                                                                                                                         rc_checkpoint, rc_tablespace, rc_datafile, rc_backup_datafile,
          [by {session | access} ] [whenever [not] successful]
                                                                                                                                          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,
v$nls_parameters, v$nls_valid_values, nls_database_parameters,
                                                                                                                                          rc_stored_script_line, rc_backup_corruption, rc_copy_corruption, rc_resync,
nls instance parameters, nls session parameters, props$
                                                                                                                                          v$backup, v$backup_set, v$backup_piece, v$backup_datafile,
Server: init<SID>.ora
                                                                                                                                          v$datafile copy, v$proxy datafile, v$offline range, v$backup redolog,
NLS LANGUAGE
                                                                                                                                          v$proxy_archivedlog, v$backup_device, v$backup_corruption,
  → NLS_DATE_LANGUAGE, NLS_SORT
                                                                                                                                          v$copy_corruption, v$backup_async_io, v$backup_sync_io,
NLS TERRITORY
  → NLS DATE FORMAT
                                                                                                                                          v$session_longops, v$session_wait
         NLS_CURRENCY (fm L), NLS_ISO_CURRENCY (fm C),
                                                                                                                                          'backup_tape_io_slaves', 'db_file_direct_io_count', 'disk_asynch_io', 'tape_asynch_io',
          NLS_DUAL_CURRENCY, NLS_UNION_CURRENCY
                                                                                                                                          'control_file_record_keep_time'
         NLS MONETARY CHARACTERS
                                                                                                                                          [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)
         NLS NUMERIC CHARACTERS (fm DG)
   → NLS_LIST_SEPARATOR
   → NLS CALENDAR
  → NLS CREDIT, NLS DEBIT
                                                                                                                                         rman [target '<user>/<pwd>@<target_db>']
lxinst [oranls=<\subseteq \text{SORA NLS33} > ] [sysdir=<path>] [destdir=<path>]
                                                                                                                                                     [catalog '<user>/<pwd>@<repos_db>' | nocatalog ]
           [help=<<u>no</u>>] [warning={0 | 1 | 2 | 3} ]
                                                                                                                                                     [auxiliary '<user>/<pwd>@<aux_db>']
lxbcnf [oranls=<\subseteq ORA NLS33 > ] [userbootdir=<path>] [destdir=<path>]
                                                                                                                                                     [ {cmdfile [=] | @} <file>] [log [=] <file> [append] ]
                                                                                                                                                     [msgno] [trace [=] '<file>'] [debug] [send [=] '<cmd>']
                                                                                                                                         set dbid [=] <target_dbid>;
               environment variables
                                                                                                                                         connect {target | catalog | auxiliary} <user>/<pwd>@<db>
NLS LANG, NLS NCHAR
                                                                                                                                          startup [nomount | mount] [force] [dba] [pfile [=] <file>];
   → NLS DATE LANGUAGE, NLS SORT
   → NLS DATE FORMAT
                                                                                                                                         shutdown [normal | transactional | immediate | abort];
          NLS_CURRENCY, NLS_ISO_CURRENCY, NLS_DUAL_CURRENCY
                                                                                                                                         {mount | open} database;
         NLS MONETARY CHARACTERS
                                                                                                                                          alter database {mount | open};
         NLS NUMERIC CHARACTERS
                                                                                                                                         host ['<cmd>'];
   → NLS_CREDIT, NLS_DEBIT
                                                                                                                                         debug {on | off}
   → NLS_COMP
                                                                                                                                         set echo {on | off};
Session:
                                                                                                                                         set command id to '<id>':
alter session set NLS_LANGUAGE=<lang> NLS_TERRITORY=<territ>;
                                                                                                                                         set snapshot controlfile name to '<new>'
                                                                                                                                          send [channel <chann> [,...] | device type <dev> [,...] ]
Package
                            DBMS_SESSION.SET_NLS(<name>,<value>)
                                                                                                                                                     '<media_man_cmd>' [parms [=] '<par>'];
SQL-Functions:
                                                                                                                                          {create | replace} script <script> {<stat>;...}
  → to_char (NLS_DATE_LANGUAGE, NLS_NUMERIC_CHARACTERS, NLS_CURRENCY, NLS_ISO_CURRENCY, NLS_CALENDAR)
                                                                                                                                         delete script <script>;
                                                                                                                                         print script <script>;
   → to_date (NLS_DATE_LANGUAGE, NLS_CALENDAR)
                                                                                                                                         run {<cmd>; ...}
   → to_number (NLS_NUMERIC_CHARACTERS, NLS_CURRENCY,
                                                                                                                                         run {execute script <script>;}
                               NLS_ISO_CURRENCY)
                                                                                                                                         sql "<stat> [' ' <file> ' '] ";
   → nls_upper (NLS_SORT)
                                                                                                                                         create catalog [tablespace <ts>];
   → nls_lower (NLS_SORT)
                                                                                                                                         upgrade catalog [tablespace '<ts>'];
  → nls initcap (NLS SORT)
                                                                                                                                         drop catalog;
  → nlssort (NLS SORT)
                                                                                                                                          register database;
                                                                                                                                         list incarnation of database;
EXPORT & IMPORT & LOADS & MIGRATION
                                                                                                                                         reset database [to incarnation <id>];
v$loadcstat, v$loadistat, v$loadpstat, v$loadtstat, dba_exp_files,
                                                                                                                                          resync catalog [from controlfilecopy ['<ctrl>']];
dba_exp_objects, dba_exp_version, sys.incexp, sys.incfil, sys.incvid
                                                                                                                                         catalog { archivelog | datafilecopy | controlfilecopy} '<file>' [,...]
                                                                                                                                                        [tag [=] '<tag>' | level [=] <X>];
         userid=<user>/<pwd> parfile=<par> file=<expdat.dmp>
                                                                                                                                          change { archivelog | datafilecopy | backuppiece | backupset | proxy
          filesize=<X> volsize=<X> log=<log> buffer=<X> silent=<N>
                                                                                                                                                        | controlfilecopy } {'<file>' | <X> | all | tag [=] '<tag>'}
          recordlength=<X> direct=<\underline{N}> rows=<\underline{Y}> indexes=<math><\underline{Y}> grants=<math><\underline{Y}>
                                                                                                                                                        { delete | available | unavailable | uncatalog | validate
          constraints=<<u>Y</u>> triggers=<Y> feedback=<<u>0</u>> inctype={complete |
                                                                                                                                                       | crosscheck };
         \begin{array}{l} \text{cumulative} \mid \text{incremental} \} \ \text{statistics} = \underbrace{\{\text{estimate}} \mid \text{compute} \mid \text{none} \} \\ \text{record} = \underbrace{Y} > \text{compress} = \underbrace{Y} > \text{consistent} = \underbrace{N} > \underbrace{\text{full} = \underbrace{N}} \mid \text{full} = \underbrace{N} > \underbrace{\text{full}} = \underbrace{N} > \underbrace{N} > \underbrace{\text{full}} = \underbrace{N} > \underbrace
                                                                                                                                         {crosscheck | delete expired} backup [ of
                                                                                                                                                      { { datafile | tablespace | database [skip tablespace] } '<name>'
          owner=<schema> | tables=(<tab>[:<part>] [,...] [query=<expr>] ) }
                                                                                                                                                      controlfile | archivelog { all | like '<name>' | {from | until}
          transport_tablespace=<<u>N</u>> tablespaces=<ts>[,...]
                                                                                                                                                      time [=] '<date>' | scn [=] <X> | logseq [=] <X> [thread = <X>] } } ]
         point_in_time_recover=<<u>N</u>> recovery_tablespaces=<ts> [,...] userid=<user>/<pwd> parfile=<par> file=<<u>expdat.dmp</u>>
                                                                                                                                                    [ {tag = '<tag>' | completed { {after | before} [=] '<date>'
                                                                                                                                                                                                   | between '<date>' and '<date>' } ];
           filesize=<X> volsize=<X> log=<log> buffer=<X> recordlength=<X>
                                                                                                                                          allocate [auxiliary] channel <chann> [for {delete | maintenance} ]
          rows = <\underline{Y}> grants = <\underline{Y}> indexes = <\underline{Y}> constraints = <\underline{Y}> commit = <\underline{N}>
                                                                                                                                                     { type [=] {disk | '<dev>'} | name [=] '<name>' }
          ignore=<\underline{N}> inctype=\{system | restore\} feedback=<\underline{0}> show=<\underline{N}>
                                                                                                                                                     [parms [=] "<par>"] [format [=] '<fm>']
          analyze=<\underline{V}> recalculate_statistics=<\underline{N}> help=<\underline{N}> destroy=<\underline{N}>
                                                                                                                                                     [connect [=] '<user>/<pwd>@<target_ops_inst>']
          skip_unusable_indexes=<<u>N</u>> indexfile=<file> toid_novalidate=
                                                                                                                                                     [debug [=] < X >] [trace [=] < X >];
          (<type>[,...]) { full=<\underline{N}> | tables=(<tab>[:<part>][,...]) }
                                                                                                                                          set limit channel <chann> [read rate [=] <X>] [kbytes [=] <X>]
          fromuser=<schema>[,...] touser=<schema>[,...]
                                                                                                                                                                                       [maxopenfiles [=] <X>];
          transport_tablespace=<<u>N</u>> datafiles='(<file> [,...] )'
                                                                                                                                          release channel [<chann>];
```

report { { need backup { {incremental | days} | redundancy} [=] <X> }

{ datafile {'<file>' | <X>} [,...]

| unrecoverable }

```
| 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>' [,...]]
       I 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>] } }
       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>}]
set until { time [=] '<date>' | scn [=] <X> | logseq [=] <X> [thread [=] <X>] };
recover { database [ until { time [=] '<date>' | scn [=] <X>
                            |\log | = | < 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> = (description_list = (description = (address_list = (address = (protocol = (address = (addres

 $\{\underline{ttc} \mid giop\}\)$ (session = $\{\underline{ns} \mid raw\}\)$))), sid_list_<LISTENER> = (sid_list = (sid_desc =

 $\langle \underline{tcp} \rangle$) (host = $\langle node \rangle$) (port = $\langle \underline{1521} \rangle$) (key = $\langle prog \rangle$))) (protocol_stack = (presentation =

| tablespace '<ts>' [,...]

```
(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 = \langle X \rangle, ssl. client authentication = \langle X \rangle, ssl. version = undetermined 1
 >> Since release 8.1 sid_list_<LISTENER> only required with enterprise manager! <<
[log_{file | directory}_{client | server} = <X>, sqlnet.expire_time = <X>, use_cman = <X>, use_dedicated_server = <X>, sqlnet.expire_time = <X>, use_cman = <X>,
{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 = \langle X \rangle, names.initial retry timeout = \langle X \rangle, names.request retries
 = \langle X \rangle names max open connections = \langle X \rangle names message pool start size = \langle X \rangle
 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} =
[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.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
 [ cprot>.(excluded | invited)_nodes = <node>, , cprot>.validnode_checking = <X>,
 tcp.nodelav = <X>1
 \begin{array}{ll} \hline [cman = (address = (protocol = < cp>) (host = < node>) (port = < 1630>)), cman_admin = (address = (protocol = < cp>) (host = < X>) (port = < 1830>)), cman_profile = (maximum_relays = < X>, relay_statistics = < X>, log_level = < X>, tracing = <
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 =
 snmp_rw.ora, services.ora, $TNS_ADMIN
            dbsnmp_status } [<<u>LISTENER</u>>]
trcasst [-o(c|d)\{u[q]|t\} - e[0|1|2] - s - p ...] < file>
Advanced Queuing / Data Warehousing
```

ckpcch.ora, sdns.ora, namesini.sql, namesupg.sql, snmp_ro.ora, Isnrctl { start | stop | status | reload | set | show | help | change_password | services | save_config | dbsnmp_start | dbsnmp_stop | 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] netasst, tnsping, trcroute, adapters, oerr <tns> <errno> Distributed DB / Replication / Heterogenous Services / 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,

tablespaces=<ts>[,...] tts_owners=<owner>[,...]

point_in_time_recover=<false>

```
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};

commit comment 'ORA-2PC-CRASH-TEST-<1-10>';

```
alter system {enable | disable} distributed recovery:
create {materialized view | snapshot} log on <tab>
       [tablespace <ts>] [storage (...) ]
       [pctfree <\underline{10}>] [pctused <\underline{40}>] [initrans <\underline{1}>] [maxtrans <X>]
        [logging | nologging] [cache | nocache] [noparallel | parallel [<X>]]
       [partition ...] [lob ...] [using index...]
       [with [primary key] [, rowid] [ (<col> [,...] ) ] ]
[ {including | <u>excluding</u>} new values];
alter {materialized view | snapshot} log on <tab>
       [add [primary key] [, rowid] [ (<col> [,...] ) ] ] [...];
drop {materialized view | snapshot} log on <tab>;
create {materialized view | snapshot} < mview>
       [tablespace <ts>] [storage (...) ]
       [pctfree <\underline{10}>] [pctused <\underline{40}>] [initrans <\underline{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>']
```

[using [default] [master | local] rollback segment [<rbs>]]]

[with {primary key | rowid}]

```
[for update]
       [ {enable | disable} query rewrite]
       as <querv>:
alter {materialized view | snapshot} < mview > ... [compile];
drop {materialized view | snapshot} < mview>;
create [force | noforce] dimension < dim>
       level </ev>
       hierachy <hier> (
            <child_lev> child of <parent_lev> [child of <parent_lev>...]
            [join key (<child_col> [,...]) references <parent_lev>] [join...])
       [attribute </ev> determines [ ( ] <dep_col> [,...) ] ] [attribute...];
alter dimension < dim>
       { add { level... | hierachy... | attribute... }
       | drop { level < lev > [restrict | cascade]
              hierachy <hier> | attribute <lev> }
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, v\$dlm_misc, v\$dlm_ress, v\$dlm_all_locks, v\$dlm_convert_local, v\$dlm_covert_remote, v\$dlm_traffic_controller, file lock, ext to obi '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>[!] [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', [<u>obsolete</u>; init_com.ora, 'gc_latches', 'ge_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)

Tuning/Contention Global cache: 'global cache %' (v\$sysstat) cache fusion latency: ~ 1-40 ms 'global cache or block receive time' / 'global cache or blocks received' 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' 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

essage statistics: (v\$dlm_misc) average receive queue length: < 10 'dlm total incoming msg queue length' / 'dlm messages received' 'DBWR forced writes' / 'physical writes' (v\$sysstat) OPS I/O ('remote instance undo header writes' + 'remote instance undo block

lock statistics: v\$dlm_convert_local, v\$dlm_convert_remote

writes') / 'DBWR forced writes' (v\$sysstat) releasable freelist waits" (v\$sysstat) Lockina

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),

Pinaina: ping write ratio: (v\$sysstat)
'DBWR cross instance writes' / 'physical writes', v\$lock_activity

v\$lock_activity, v\$class_ping, v\$ping

Block content: v\$bh, v\$cache, v\$ping mult. copies of 2nd block of file -> freelist contention (check v\$waitstat) 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 Engu Locks: 20 + (10*sess) + db_files + 1 + (2*proc) + (db_block_buffers/64)

DMI Locks: set 'dml_locks' = 0 for all instances, or disable specific table locks '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

IDLM:

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],
sufffix], 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], I[ist] [<X>] [<Y>] [*] [last], i[nput]
@<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>]
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>] | Ino | 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] | reph[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[i] <string>]
         [fold_a[fter] | fold_b[efore] ] [like <alias>] [newl[ine] ]
         [ {new_v[alue] | old_v[alue] } <var>]
         [ jus[tify] { I[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
                    (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 }

```
{-2.147.483.647 .. 2.147.483.647} library arithmetic
   binary integer
                         (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(\langle p \rangle, \langle s \rangle) {38digits = 21B (20B Mantisse, 1B Exponent)}
                         (Subtypes: dec, decimal, double precision, float, int,
                        integer, numeric, real, smallint)
                        {7B=CentYearMonDayHourMinSec, -4.712 to 9.999}
                        {pl: 32.767B, col: 2.000B}
                        {ext: 10B, restr. 6B}
   rowid
                        col: urowid [ (<4000B>) ] {physical and logical rowids}
                        long {pl: 32.760B, col: 2^31-1B=2G},
    large objects:
                        long raw {pl: 32.760B, col: 2^31-1B=2G}
                        internal: BLOB, CLOB, NCLOB (2^32-1B=4G)
                                             BFILE(pointer)
                                  create [or replace] directory <dir> as '<path>';
                                  drop directory <dir>;
                        {true | false | null}
   boolean
subtype <subtype> is <base type> [not null]
                        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>(...)
   varrav
                        may be DB col
   nested table
   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>] ) ], <math>exists(<i>), extend [(<n>,(<i>] )], <math>limit, first, last,
        next(\langle i \rangle), prior(\langle i \rangle), trim[(\langle i \rangle)]
                         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		Х		Х	X	Х			X
char	Х		X	Х	Х	Х	Х	X	Х
date		Х		Х					X
long		Х					Х		X
number	Х	Х		Х		Х			Х
pls_int	Х	Х		Х	Х				Х
raw		Х		Х					X
urowid		Х							Х
vorobor?		V	V	· ·		· ·	· ·	V	

| never refresh }] 25-Jan-2002

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

	char, varchar2	number	date	raw	rowid, urowid	nchar, nvarchar2
char, varchar2	X	Х	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 [>> 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

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 Itrim, rtrim, Ipad, rpad, replace, translate, initcap, concat, chr, nls_upper, nls_lower, nlssort, nls_initcap, soundex, sysdate, next_day, last_day, add_months, months_betwee new time, ref, deref, make ref, reftohex, value, multiset, cast, level, prior

<fm>= 90\$BMISPRDGCL,.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}])

{ <WA-Func> | ratio to report } (<col>)

over ([partition by <col> [,...]] [asc | desc] [nulls {first | last}])

{lag | lead} (<col>, <default>)

over (order by <col> [,...] [asc | desc] [nulls {first | last}])

alter, create, drop, audit, noaudit, grant, revoke, update, truncate,

select, insert, delete, lock table, explain plan

TxCtl commit, rollback, savepoint, set transaction

SessCtl alter session, set role

SysCtl alter system

Access Paths single row by rowic

single row by cluster join

single row by hash cluster key with unique or primary key single row by unique or primary key

cluster ioin hash cluster key

indexed cluster key composite key single-column indexes

bounded range search on indexed columns unbounded range search on indexed columns

sort-merge join max or min of indexed column

25-Jan-2002

order by on indexed columns full table scan

sample table scan fast full index scan

index join

bitmap index scan {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 (ctab), Finds (_dtab), Finds _A, Finds _B, INDEX (ctab) = (ands [_i]), INDEX_DESC (ctab) = (inds [_i]), INDEX_DESC (ctab) = (inds [_i]), INDEX_DESC (ctab) = (inds [_i]), INDEX_DESC (_dtab) = (inds [_i]), INDEX_ STAR, USE_NL (<tab>[...]), USE_MERGE (<tab>[...]), USE_HASH (<tab>[...]), DRIVING SITE (<tab>[...]), PARALLEL (<tab>[...]), NOPARALLEL (<tab>[...]), PQ_DISTRIBUTE (<tab>[,] <out>, <in>), APPEND, NOAPPEND, PARALLEL_INDEX (tab">tab">tab [...] (tab [...] (tab [...]), (tab [...]), NOARALLEL_INDEX (tab [...]), NOACHE (<a href="ta 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> <querv>: select { [aggr_func (] [distinct | unique | all] { [<alias>.]<col> | * } [)] [{+ | - | * | /} <expr>] [as] ["<alias>"] [,...] | <seq>.{nextval | currval} cursor (<subquery>) } [[bulk collect] into <var> [,...]] [sample [block] (<X>)] | (<subquery> [with {read only | check option [constraint < constr>] }]) | table (<coll_expr>) [(+)] } where [(] [(] [< alias > .] < col/expr > [(+)] [, < expr > ...)] $\{\{ = | != | \sim = | <> | <= | >= | <| > \} [any | some | all]$ { (<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 $\langle tab \rangle$ set $\langle col \rangle = \{\langle val \rangle \mid \langle string \rangle \} [,...];$ delete [from] <tab> [partition (<part>)] [alias] [where <expr>]; rollback [work] [to [savepoint] <sp> | force '<id>']; set transaction {read only | read write isolation level (serializable | read committed) use rollback segment <rbs>};

commit [work] [comment '<string>' | force '<id>' [,<scn>]]; savepoint <sp>

alter session {enable | disable} commit in procedure;

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

-- <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 [<< <bl/>blocklabel> >>]

if <expr> then <stat>; [elsif <expr> then <stat>;]

[else <stat>;] end if; [<< <label> >>]

[exit [<label>] when <expr>;] end loop [<label>]; for all $\langle i \rangle$ in $\langle a \rangle ... \langle b \rangle$ <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>}; open <refcurs> for <query> [using <var> [,...]]; execute immediate '<dyn_sql>' [into {<var> [,...] | <rec>}] [using [in | out | in out] < arg > [,...]]; 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> [,...]]; 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>];

<stat>; [sqlcode; sqlerrm(<X>);]

when {<exc> | others} [or <exc2> ...] then

>> Predefined Server Exceptions:
no_data_found, too_many_rows, invalid_cursor, zero_divide, dup_val_on_index <<

I while <expr>

<stat>

| for <i> in [reverse] <a>...

[if <expr> then exit;]

| for <rec> in {<curs> [(<par>,...)] | (<query>) }]

return [[(] <expr> [)]];

AND true false null OR true false null NOT true true false null true true true true false false false false false true false null 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}]

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 create [(<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 cproc>; 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> | - }) [...]) 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 <tab> table (<col> <type> [default <expr> [null | not null] | not null [with default]]...); exec sql [for <:var>] allocate descriptor

exec sql declare <stat> statement;

[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>}];

Page 5

```
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> };
#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}
sess_sh -h[elp] -version -d[escribe] -role <role> -user <user>
          -password <pwd> -service <url> -{ssl | iiop}
deployejb -generated <cli>entjar> -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
```

25-Jan-2002