


```

| 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 (<lobst> [(tablespace <ts>)] ) ]
[nested table <col> store as <tab> [ ( (<prop>) [storage (...)] ) ]
  [return as (locator | value)]
[on commit {delete | preserve} rows];
create table <tab> [logging | nologging] ... as select...;
alter table <tab> modify (<col> <type>...);
alter table <tab> add (<col> <type>...);
alter table <tab> set unused { (<col> [...] ) | column <col> }
  [cascade constraints] [invalidate];
alter table <tab> drop { (<col> [...] ) | column <col> }
  [cascade constraints] [invalidate] [checkpoint <512>];
alter table <tab> drop (unused columns | columns [continue]
  [checkpoint <512>]);
drop table <tab> [cascade constraints];
rename <tab> to <new_tab>;
alter table <tab> move [tablespace <ts>] [storage (...)]
  [logging | nologging] [noparallel | parallel [<X>]];
truncate table <tab> [ {preserve | purge} snapshot log]
  [ {drop | reuse} storage];
alter table <tab> [storage (...)] [noparallel | parallel [<X>]] ...
  [ {nominiimize | minimize} records_per_block];
alter table <tab> { allocate extent
  ( [size <X>] [datafile '<files>'] [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> drop partition [<hash_part> [tablespace <ts>]];
  | alter table <tab> add partition <part> [...] ;
  | alter table <tab> coalesce partition;
  | alter table <tab> truncate (partition | subpartition) <part>
    [ {drop | reuse} storage];
  | alter table <tab> rename (partition | subpartition) <part> to <new>;
  | alter table <tab> modify (partition | subpartition) <part>
    [storage (...)] [allocate extent... [logging | nologging] ...
    [ {rebuild} unusable local indexes]
    [ 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_constraint>]
    { 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> ... [relly | norely];
  | alter table <tab> drop
    { constraint <constr> | primary key | unique (<col> [...] ) } [cascade];
  | set constraint[s] <constr> [...] | all | immediate | deferred;
  | alter table <tab> {enable | disable} all triggers;
  | create [or replace] trigger <trigg> [before | after | instead of]
    { {delete | insert | update [of <col> [...] ] | [or... ]
    on <tab> | [nested table <col> of] <view> }
    | { create | alter | drop } [or... ]

```

```

| {shutdown | startup | servererror | logon | logoff} [or... ]
on (schema | database) }
referencing
  {old [as] <old> | new [as] <new> | parent [as] <parent> } [...] ]
[for each row] [when (<expr> ) ]
{ begin <stat>; end;
| call ... ; }
alter trigger <trigg> { enable | disable | compile [debug] };
drop trigger <trigg>;
analyze table <tab> (<jobst> <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>, <networks> ]
  [default selectivity <selc>];
disassociate statistics from
  { columns [<tab>]<col> [...]
  | functions <func> [...] | packages <pack> [...] | types <type> [...]
  | indexes <ind> [...] | indextypes <indtype> [...] } [force];

```

IEWS & SYNONYMS & SEQUENCES

```

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

```

CLUSTERS

```

dba_clusters, dba_clu_columns, all_tab_columns,
dba_rebuild_hash_expressions
create cluster <clus> (<col> <type> [...] )
[ {index | [single table] hashkeys <X> [hash is <expr> ] ]
[size <1KBS>] [tablespace <ts>] [storage <X> ]
[ {pctfree <10> } [pctused <40>] [initrans <X>] [maxtrans <255>];
create index <ind> on cluster <clus>
[storage (...)] [tablespace <ts>] [pctfree <X> ]
[initrans <X>] [maxtrans <X>];
create table <tab>
(<col> <type>... [constraint <constr>...] )
cluster <clus> (<col> [...] );
alter cluster <clus> ...;
truncate cluster <clus> [ {drop | reuse} storage];
drop cluster <clus> [including tables [cascade constraints] ];
analyze cluster <clus> ...;

```

INDEX-ORGANIZED TABLES

```

all_tables (iot_type, iot_name), all_indexes
create table <iot> (<col>... primary key...)
organization index
[tablespace <ts>] [pctfree <X>] [initrans <X>] [maxtrans <X>]
[storage (...)] [pctthreshold <50>] [including <col>] ]
[compress [<X>] | nocompress]
[ overflow [tablespace <ts>] [pctfree <10>]
  [initrans <1>] [maxtrans <255>] [storage (...)]
  [allocate... ] [deallocate... ] [logging | nologging] ]
[ partition by range (<col> [...] )
  ( partition <partX> values less than (<value> [...] )
  [storage (...)] [tablespace <ts>] [overflow tablespace <ts>...]
  [ , partition... ] );
alter table <iot> ... [overflow...];
alter table <iot> add overflow ... [ (partition <part>... )];

```

```

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

```

INDEXES

```

dba_indexes, dba_indextypes, dba_indextype_operators, dba_ind_columns,
dba_ind_expressions, index_stats, dba_part_indexes, dba_ind_partitions,
dba_ind_subpartitions, dba_part_col_statistics, dba_subpart_col_statistics,
index_histogram
'create_bitmap_area_size', 'bitmap_merge_area_size'
Package DBMS_PCLXUTIL (build_part_index)
Tuning
(index_stats) del_if_rows_len / lf_rows_len > 20% -> rebuild index

```

```

create [unique | bitmap] index <ind>
on <tab> [ (<expr> ) <col> [desc] [...] ]
[tablespace <ts> | default ] [storage (...)]
[ {pctfree <10> } [initrans <X>] [maxtrans <255>]
[logging | nologging] [nosort] [reverse] [online]
[noparallel | parallel [<X>]] [nocompress | compress [<X>]]
[ local
  { { partition [<partX>] [storage (...)] [tablespace <ts>]
  [logging | nologging] [ , partition... ] ]
  | { store in { [<ts> [...] ] default }
  [ 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 ( (<col> [...] ) [maxvalue] )
  [storage (...)] [tablespace <ts>] [logging | nologging]
  [ , partition... ] ) ]
[ indextype is <type> [parameters ('<str>') ] ];
drop index <ind>;
alter index <ind> {enable | disable};
alter index <ind> unusable;
alter index <ind> rename to <new_ind>;
alter index <ind> drop partition <part> [...] ;
alter index <ind> rename (partition | subpartition) <part> to <new>;
alter index <ind> modify (partition | subpartition) <part>
[storage (...)] ... [logging | nologging] [unusable]
[rebuild unusable local indexes];
alter index <ind> modify default attributes [for partition <part>]
[storage (...)] [pctfree <X>] ...;
alter index <ind> rebuild (partition | subpartition) <part>
[tablespace <ts>] [parallel [<X>]];
alter index <ind> split partition <part1> at values less than (<X>)
into ( partition <part2>, partition <part3> [...] );
alter index <ind> [storage (...)] [initrans <X>] [maxtrans <X>]
[nocompress | compress <X>];
alter index <ind> allocate extent
( [size <X>] [datafile '<file>'] [unusned <X>] );
alter index <ind> [datafile [datafile]] deallocate unused [keep <X>];
alter index <ind> rebuild
[ { partition | subpartition } <part> ] [tablespace <ts>] [storage (...)]
[ {pctfree <10> } [initrans <X>] [maxtrans <255>]
[logging | nologging] [noparallel | parallel [<X>]]
[nocompress | compress <X>] [compute statistics] [online]
[noreverse | reverse] [parameters ('<par>') ];
alter index <ind> coalesce;
analyze index <ind> ...;
analyze index <ind> validate structure;

```

ROLLBACK SEGMENTS

```

v$rollname, v$rollstat, v$transaction, v$transaction_enqueue,
v$global_transaction, dba_rollback_segs, dba_pending_transactions
'rollback_segments', 'transactions', 'transactions_per_rollback_segment'
Package DBMS_TRANSACTION (use_rollback_segment)
Tuning/Contention
RBS Header: "undo segment tx slot" (v$sysstat.event) > 0 or
(v$rollstat) sum(waits) / sum(gets) > 5% or
-> add RBS
RBS Segment: "%undo0n" (v$waitstat) / "consistent gets" (v$sysstat) (count/value) > 1%
-> add RBS

```

```

create [public] rollback segment <rs> [tablespace <ts>]
[storage ( [initial <5xBS>] [next <5xBS>] [optimal <null>]

```

```

[ minextents <1> ] [ maxextents (<X> | unlimited) ] );
drop rollback segment <rs>;
alter rollback segment <rs> [online | offline];
alter rollback segment <rs> storage (...);
alter rollback segment <rs> shrink [to <X>];
set transaction use rollback segment <rs>;

```

TEMPORARY SEGMENTS

```

v$sort_segment, v$sort_usage, dba_segments
'sort_area_size', 'sort_area_retained_size', 'sort_multiblock_read_count'
'lobseize', 'sort_direct_writers', 'sort_write_buffers', 'sort_write_buffer_size'
Tuning
'sorts (disk)', 'sorts (memory)', 'sorts (rows)' (v$sysstat)
disk.value / mem.value > 5%
-> increase 'sort_area_size' (+ decrease 'sort_area_retained_size')

```

USERS & PRIVILEGES & RESOURCES & POLICIES

```

v$enableprivs, v$resresource, v$resource_limit, v$pwfile_users, v$context,
v$srcr_plan, v$srcr_plan_cpu_mth, v$srcr_consumer_group,
v$srcr_consumer_group_cpu_mth, v$parallel_degree_limit_mth,
v$max_active_sess_target, dba_users, dba_roles, dba_profiles,
dba_ustats, dba_ts_quotes, dba_sys_privs, dba_tab_privs, dba_col_privs,
dba_role_privs, role_sys_privs, role_tab_privs, role_role_privs,
user_tab_privs_recd, user_tab_privs_recd, user_col_privs_made,
user_col_privs_read, user_password_limit, 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
'ot_dictionary_accessibility', 'remote_os_authent', 'os_roles', 'remote_os_roles',
'max_enabled_roles', 'resource_limit', 'resource_manager_plan', 'ent_domain_name'
Environment: SORA_ENCRYPT_LOGIN
Packages
DBMS_RESOURCE_MANAGER (set_initial_consumer_group, (create |
submit | clear | validate)_pending_area, (create | update | delete)_plan |
plan_directive | consumer_group), delete_plan_cascade,
switch_consumer_group_for_(sess | user) ),
DBMS_RESOURCE_MANAGER_PRIVS ( (grant | revoke)_system_privilege,
(grant | revoke)_switch_consumer_group), DBMS_SESSION
(switch_current_consumer_group), DBMS_RLS (add | drop | enable |
refresh_policy)

```

```

create user <user>
identified { by <pwd> | by values '<crypt_pw>'
| externally | globally as '<user>' }
[ default tablespace <ts> ] [temporary tablespace <ts>]
[ quota <X> | unlimited on <ts> [ quota... ] ]
[ password expire ] [account {lock | unlock} ]
[ profile {<prof> | default} ];
alter user <user> ...;
drop user <user> [cascade];
create role <role> [ {not} identified {by <pwd> | externally | globally} ];
alter role <role> ...;
drop role <role>;
alter user <user> default role (<role> [...] ) | all [except <role> [...] ] | none;
set role { <role> [identified by <pwd>] | <role> [identified by <pwd>]... ]
| all [except <role> [...] ] | none ;
grant { <priv> [...] | <role> [...] } to
{ <user> [...] | <role> [...] } [ public ] [with admin option];
revoke (<priv> | <role>) from { <user> | <role> | public };
grant { <priv> [ (<col> [...] ) ] [...] | all on <object>
to { <user> [...] | <role> [...] } [ public ] [with grant option];
revoke { <priv> [ (<col> [...] ) ] | all [privileges] } on [directory] <object>
from { <user> | <role> | public } [cascade constraints];
create profile <prof> limit
(sessions_per_user <X> | unlimited | default] ]
(cpu_per_session <X> | unlimited | default] ]
(cpu_per_call <X> | unlimited | default] ]
[connect_time <X> | unlimited | default] ]
[idle_time <X> | unlimited | default] ]
[logical_reads_per_session <X> | unlimited | default] ]
[logical_reads_per_call <X> | unlimited | default] ]
[composite_limit <X> | unlimited | default] ]
[private_sga <X> | unlimited | default] ]
[failed_login_attempts <X> | unlimited | default] ]
[password_lock_time <X> | unlimited | default] ]
[password_life_time <X> | unlimited | default] ]
[password_grace_time <X> | unlimited | default] ]
[password_reuse_time <X> | unlimited | default] ]
[password_verify_max <X> | unlimited | default] ]
[password_reuse_function (<func> | null | default] );
alter profile <prof> limit...;
drop profile <prof> [cascade];
alter resource cost [connect_time <X>] [cpu_per_session <X>]

```



```

dba_mvview_analysis, dba_mvview_detail_relations, dba_summaries,
dba_summary_aggregates, dba_summary_joins, dba_summary_keys,
dba_summary_detail_tables, dba_dimensions, dba_dim_levels,
dba_dim_hierarchies, dba_dim_level_child_of, dba_dim_attributes,
dba_dim_join_key, dba_dim_level_key, mvview$_exceptions,
mvviews$_recommendations, mvview$_evaluations, hs_all_caps,
hs_class_caps, hs_base_caps, hs_inst_caps, hs_all_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_relocking', 'hs_fds_fetch_rows',
'hs_rpc_fetch_size'
[obsolete: default, 'distributed_lock_timeout', 'snapshot_refresh_keep_connections',
'snapshot_refresh_processes', 'snapshot_refresh_bsp',
'distributed_recovery_connection_hold_time', 'job_queue_keep_connections']

```

Packages

```

DBMS_RECAT ( (create | drop). master_regroup, {suspend |
resume}_master_activity, {create | drop}_master_reobject, set_columns, {add |
remove}_master_database, alter_master_propagation, relocate_masterdef,
{make | drop}_column_group, {add | drop}_group_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_reobject, switch_snapshot_master,
send_and_compare_old_values, {register | cancel | purge}_statistics,
do_deferred_recat_admin, purge_master_log, recat_import_check,
comment_on_<regroup | reobject | repsites | column_group | priority_group |
site_priority | unique_resolution | update_resolution | delete_resolution>,
DBMS_RECAT_ADMIN {grant_admin | schema | any_schema},
register_user_regroup(), DBMS_RECAT_INSTANTIATE,
DBMS_RECAT_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, register_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_tam_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_REVIEWER_DIFF
(differences, rectify, ), DBMS_AQ, DBMS_AQADM, DBMS_MVIEW {refresh,
refresh_all_mvviews, refresh_dependent}, DBMS_OLAP (validate, dimension,
estimate_space, recommend_mv, estimate_summary_size, evaluate_utilization,
evaluate_utilization_v, set_logfile_name), DEMO_DIM (print_dim, print_allinds),
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, catrep.sql, smdim.sql, sadvdemo.sql, caths.sql

```

```

create {shared | public} database link <link>[@qual]
[connect to {<user> identified by <pwd> | current_user}]
[authenticating by <user> identified by <pwd>]
[using '<netstring>'];
alter session close database link <link>;
drop {public} database link <link>;
alter session enable {commit | rollback | nothing};
alter system {enable | disable} distributed recovery;
commit comment 'ORA-2PC-CRASH-TEST-41-10';

```

```

create {materialized view | snapshot} log on <tab>
[tablesapce <ts>] [storage (...)]
[pcftree <10>] [pctused <40>] [initrans <1>] [maxtrans <X>]
[logging | nologging] [cache | nocache] [nonparallel | parallel [<X>]]
[partition ...] [lob ...] [using index... ]
[with {primary key | rowid} [ , rowid] [ , <col>] [ , ... ] ]
[ { including | excluding} new values];
alter {materialized view | snapshot} log on <tab>
[add {primary key | rowid} [ , rowid] [ , <col>] [ , ... ] ] [ , ...];
drop {materialized view | snapshot} log on <tab>;
create {materialized view | snapshot} <mvview>
[tablesapce <ts>] [storage (...)]
[pcftree <10>] [pctused <40>] [initrans <1>] [maxtrans <X>]
[logging | nologging] [cache | nocache] [nonparallel | parallel [<X>]]
[cluster <clust> <col>] [ , ... ] ] [lob... ] [partition... ]
[build {immediate | deferred}]
[on prebuilt table [ { with | without} reduced precision]]
[using index... ]
{ [refresh {fast | complete | force} [on commit | on demand]
[start with <date>] [next <date>]
[with {primary key | rowid}]
[using {default} [master | local] rollback segment [<rbs>]] ]
| never refresh ]}

```

```

[for update]
[ {enable | disable} query rewrite]
as <query>;
alter {materialized view | snapshot} <mvview> ... [compile];
drop {materialized view | snapshot} <mvview>;
create {force | noforce} dimension <dim>
level <lev>= [ ( [ <tab>. <col>] [ , ... ] )] [level... ]
hierarchy <hier> (
  <child_lev> <child_of_<parent_lev>... ]
  [join key (<lev> | <parent_lev>)] references <parent_lev> [join... ] ]
[attribute <lev> determines [ ( [ <dep_col>] [ , ... ] )] [attribute... ]];
alter dimension <dim>
{ add [ level... | hierarchy... | attribute... ]
| drop [ level <lev>] [restrict | cascade]
| hierarchy <hier> [ attribute <lev>]
| compile };
drop dimension <dim>;

```

Parallel Server

```

gv$<DYN_PERF_VIEW>, v$active_instances, v$resource, v$resource_limit,
v$ping, v$class_ping, v$file_ping, v$temp_ping, v$false_ping,
v$lock_activity, v$lock_element, v$locks_with_collisions, v$lock_activity,
v$lock_class_ping, v$cache_lock, v$dlm_latch, v$latch_misses,
v$dlm_locks, v$dlm_misc, v$dlm_revs, v$dlm_all_locks,
v$dlm_convert_local, v$dlm_convert_remote, v$dlm_traffic_controller,
file_lock_ext_to_obj
'parallel_server', 'parallel_server_instances', 'thread', 'cpu_count', 'instance_number',
'instance_groups', 'parallel_instance_group', 'service_names', 'dml_locks',
'gc_instances_to_locks' = '<#>[ -<#>] = <X>[ <L>B] [ ] [each] [ : ... ]', 'gc_releasable_locks',
'gc_rollback_locks', 'gc_defer_time', 'gc_latches', 'lms_locks', 'lms_res', 'lms_procs',
'max_commit_propagation_delay', 'parallel_default_max_scans', 'lock_name_space',
[obsolete: init_com.ora, gc_latches', 'gc_locks', 'delayed_logging_block_cleanouts',
'freeze_db_for_fast_instance_recovery', 'lms_home', 'ops_admin_group']
init_db_name=ora, <db_name>.conf, IDLM,PCM, OPQ, OPSM, OPSP
Background: LMON, LMD0, LCK<#>, BSP<#>
vendor OSDs: CM, Start, IO, IPC (RegKeys: CMDLL, IODLL, IPCDLL, STARTDLL)
catpar.sql

```

Tuning/Contention

```

Global cache: 'global cache %' (v$sysstat)
cache fusion latency: ~ 140 ms
'global cache or block receive time' / 'global cache or blocks received'
consistent-read requests:
'global cache or block received' + 'global cache or blocks read from disk'
average get time: ~ 20-30 ms
'global cache get time' / 'global cache gets'
average convert time: ~ 10-20 ms
'global cache convert time' / 'global cache converts'
timeouts:
'global cache or timeouts', 'global cache convert timeouts' = 0
Global locks:
average global lock gets: ~ 20-30 ms
'global lock get time' / 'global lock sync gets' + 'global lock async gets'
average global lock convert time: ~ 20 ms
'global lock convert time' /
'global lock sync converts' + 'global lock async converts'
IDLM:
DLM non-PCM resources: 'V$library_cache', v$rowcache
lock statistics: v$dlm_convert_local, v$dlm_convert_remote
message statistics: (v$dlm_misc)
average incoming msg length:
dlm total receiving queue length / 'dlm messages received'
OPS I/O:
'DBWR forced writes' / 'physical writes' (v$sysstat)
('remote instance undo header writes' + 'remote instance undo block
writes') / 'DBWR forced writes' (v$sysstat)
Locking:
'releasable freelist waits' (v$sysstat)
Lock conv.:
lock hit ratio: (v$sysstat)
consistent gets' - 'global lock converts (async)' / 'consistent gets' > 95%,
'lock element cleanup' (V$system_event, v$session_wait),
v$lock_activity, v$class_ping, v$ping
Pinging:
'ping write ratio: (v$sysstat)
'DBWR cross instance writes' / 'physical writes', v$lock_activity
Block content:
v$bh, v$cache, v$ping
mult. copies of 2nd block of file -> freelist contention (check v$waitstat)
partition tables and indexes OR
configure process free lists and free list groups + allocate extents for
instances (free list group choice, 'alter session set instance = <X>';)
PCM Locks:
'lms_locks' = 'lms_res' = 2 * 'gc_files_to_locks + gc_rollback_locks/fixcd'
+ gc_releasable_locks, v$resource_limit,
20 + (10*sess) + db_files + 1 + (2*proc) + (db_block_buffers/64)
Enqu. Locks:
set 'dml_locks' = 0 for all instances, or disable specific table locks
DML Locks:
'Recovery: 'instance recovery database freeze count' (v$sysstat)
Inst. groups
'alter session set parallel_instance_group = <grp>;'

```

```

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

```

Fail Safe

```

fscmd { dumpcluster | movegroup | onlinergroup | 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

```

app[ro]nto, array[size], auto[commit], auto[print], auto[recovery], auto[trace] [trace[only]
expl[ain] stat[istics], bio[compatibility], c[on]ds[cripts], colsep, col[padding], col[sep],
copy[on]mit, copy[ty]pecheck, def[ine], describe [depth | indent | linenum], echo, edit[file],
end[ded]ed, esc[ape], feed[back], flagger, flu[sh], head[ing], head[sep], instance, init[size],
loboff[set], logsource, long, long[chunksize], newpage[page], num, num[format], num[width],
pagesize[set], pause[se], recsep, recsepchar, serverout[put], shift[out], show[mode],
sql[anklines], sql[case], sql[con]vert, sql[un]buffer, sql[pre]fix, sql[pr]ompt, sql[term]inator,
suffix[tab, termout], time, timing, trim[off], trim[spool], undef[erline], verify, wrap]
sql.pno, sql.lno, sql.release, sql.sqlcode, sql.user

```

```

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

```

```

/, /run]
@ <file>, @ @ <file>, start <file>
save[ ] <file> [ create ] | rep[lace] | app[end] ]
get <file> [ list ] [ no[ list] ]
spo[ol] <file> [ off ] [ off ]
[hit] [<var>], help, rem[ark], set
[no] [st] [ ! ] [ <command>
store [set] <file> [ create ] | rep[lace] | app[end] ]
def[ine] <var> = <value>, undef[ine] <var>
pro[mp] [ <string> ], pau[se] [ <string> ]
connect [ internal] [ as (sysdba | sysoper) ], disconnect
pass[word] [ <user> ]
reco[ver]

```

```

attribute <object_type> <attr> [ all[as] <name>] [ for[mat] <fm>] [ like <attr> ]
[ clear ] [ on ] [ off ]
acc[ept] <prompt> [ num[ber] ] [ char | date] [ for[mat] <fmt>] [ def[ault] <def> ]
[ <var>] [ <string>] [ no[ r] [ate] ] [ hide ]
desc[ribe] { <tab> | <view> | <pack> | <func> | <proc> | <syn> | <type> }
sho[w] <var> [ all | errors] [ { package | package body | function
| procedure | view | trigger | type | type body} <name>] [ lno | pno |
user | title] [ bit[itle] | rep[header] | rep[footer] | spool] [ | sqlcode
| sga | parameters | release]
time[ng] [start <string>] show [ stop ]
exec[ute] [ <var> = <func> <par> [ , ... ] ]
whenever {sqlerror | oserror} {exit... | continue [commit | rollback | none]}
{exit | quit} [ success ] [ failure | warning | <X> | <var> | <var> ]
[ commit ] [ rollback ]

```

```

[ title] [ bit[itle] | rep[header] | rep[ooter] ]
[ { [page] | left] | center] | right] [ col <X>] [ tab <X>] [ bold ]
[ s[kip] <X>] [ format <fm>] [ <string>] [ <var>] [ , ... ] [ on ] [ off ] ]
col[umn] [ <col>
{ [ for[mat] <fm> ] [ wrapped] | wor[d_wrapped] | tru[n]cated ]
[ heading <string>] [ all[as] | like] [ col <X>] [ null] [ <string> ]
[ fold_a[fter] | fold_b[e]fore ] [ alias <alias>] [ new[line] ]
[ { new_v[alue] | old_v[alue] } <var> ]
[ justify ] [ left] | center] | right] [ right] ]
[ on ] [ off ] [ print ] [ noprint] | clear ] ]
br[e]ak [ on { <bcol> } row | report | <expr>] [ skip] <X> | page] [ on... ]
[ n[on]duplicate ] [ duplicate ] ]
comp[ute] [ { sum | min[imum] | max[imum] | avg | std | var[iance]
| cou[nt] | num[ber] ] [ , ... ] [ label <lab> ]
of <col> <col>... ] on { <bcol> } row | report ]
clear [ scr[reen] | col[umns] | br[e]aks ] [ comp[utes] ] [ sq ] [ time[ng] | buff[er] ] ]
copy [ from <user> @ <db> ] [ to <user> @ <db> ]
{ create | replace | insert | append } <tab> [ ( <col>, ... ) ]
using <query>;

```

Data Types SQL*Plus

```

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

```

DATA TYPES (PL/SQL & DB columns)

```

v$type_size, v$temporary_objects, dba_types, dba_type_attrs,
dba_type_methods, dba_col_types, dba_lobs, dba_part_lobs,
dba_lob_partitions, dba_lob_subpartitions, dba_varrays, dba_rels,
dba_operators, dba_oparuments, dba_opbindings, dba_opciliary,
dba_method_params, dba_method_results, dba_directories, dba_rulesets

```

Scalar:

```

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

```

```

binary_integer {-2.147.483.647 .. 2.147.483.647} library arithmetic
(Subtypes: natural (non-not null), natural (not null)
[positive] (pos.), positive (non-not null), signtype(-1,0,1) )
pls_integer {38digits = 21B (20B Mantissa, 1B Exponent) }
(Subtypes: dec, decimal, double precision, float, int,
integer, numeric, real, smallint)
{ 7B = CentYear, MonDayHourMinSec, -4.712 to 9.999 }
date {pt. 32.767B, col: 2.000B}
rowid
{ ext: 10B, restr. 6B }
col: urowid ( (<4000B> )) { physical and logical rowids }
large objects:
long (pt. 32.760B, col: 2^31-1B=2G),
long raw (pt. 32.760B, col: 2^31-1B=2G)
internal: BLOB, CLOB, NCLOB (2^32-1B=4G)
external:
BFILE {pointer}
create [or replace] directory <dir> as '<path>';
drop directory <dir>;
boolean {true | false | null}
subtype <subtype> is <base_type> [not null];

```

Relationship:

```

ref ref cursor, ref <otype> {pointer}

```

Record:

```

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

```

Collection:

```

elements of same type, initialized by constructor <collect> (... )
varray may be DB col
nested table may be DB col
index-by table may not be DB col
type <varr_type> is varray [by array] (<size> of <type> [not null];
type <tab_type> is table of <type> [not null] [index by binary_integer];
<col> { <varr_type> | <tab_type> };
<col> (<subscript>)[ , <item> ] := <expr>;
<col> . <method>:
count, delete [ (<b> | <c> ) ], exists (<id>), extend [ (<n> | <b> ) ], limit, first, last,
next (<id>), prior (<id>), trim [ (<b> ) ]

```

User-defined:

```

abstract types initialized by constructor <type> (... )
create [or replace] type <type>;
forward type definition / incomplete type
create [or replace] type <type> [authid (current_user | definer) ] [is | as]
{ object (<attr> <type> [ , ... ]
[ , { static | [map | order] member } (function | procedure) <func>
[ ( { [self | <par>] [in | out] in out] <type> [ , ... ] )] [return <type> ]
[ (is | as) language { java name '<func>' | C [name <func>]
| library <lib> [with context] [parameters (<par> ) ] ]
[ , pragma restrict_references
( { <methods> | default, {rnds | wnds | rmps | 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
| asin
| 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
( { <methods> | default, {rnds | wnds | rmps | wnps | trust } ) ]
[ , ... ] } } }

```

drop type {body} <type> {force};

```

[ref] obj_type, type, varchar2(x), number(p,s), date, raw(x), character(x), char varying(x),
varchar(x), numeric(p,s), decimal[[p,s]], integer, smallint, float(x)), double precision,
real, blob, clob, bfile

```

Implicit Conversions

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


```

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 | insensitive | insensitive} ]
        [holdability = {true | false} ] [returnability = {true | false} ]
        [updatecolumns = '<col>' [...]' ] [<var> = <val>' [...]' ] )
  (<type> [<col>] [...]);
  named or positional iterator
#sql <mod> context <cont> [implements <intfc> [...]]
  [with (... <var>=<val>' [...]' )];
#sql [ [<conn_cont_inst>, <exec_cont_inst> ]
  [<var / iter> =] { <SQL stat> };
  >> Curly braces are part of syntax! <<
#sql { select /*+ <HINT> */ <expr> [... ] into <:[out] var> [... ]
  from <tab> [where <expr> ... ] };
#sql <iter> = { select <expr> [... ] from <tab> [where <expr> ... ] };
#sql { fetch <iter> into <:var> [... ] }; <iter>.next(), <iter>.endFetch(), <iter>.close()
#sql { insert into... };
#sql { update... };
#sql { delete from... };
#sql { commit };
#sql { rollback };
#sql { set transaction <mode> [ , isolation level <level> ] };
#sql { call <proc> (<par> [... ] ) };
#sql <var / iter> = { values ( <func> (<par> [... ] ) ) };
#sql { set <:var> = <expr> };
#sql <iter> = { cast <:result_set> };
#sql { [declare <var> <type>:] begin <stat> [... ] end; };
-----
sqlj -d[ir]=<dir> -encoding=<enc> -url=<url> -status -compile=false
  -user=<user>/<pwd>@jdbc:oracle:thin@<host>:<port>:<sid>
  -linemap -profile=false -ser2class -P<opt> -C<opt> -P<opt> -P<opt> -P<opt>
  -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 -f[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 -f[ecurse] -h[elp] -version -d[escribe] -role <role>
  -user <user> -password <pwd> -service <url> -{ssl | iiop}
  <name>
sess_sh -h[elp] -version -d[escribe] -role <role> -user <user>
  -password <pwd> -service <url> -{ssl | iiop}
deployejb -generated <clientjar> -descriptor <file> -verbose -republish
  -beanonly -addclasspath <path> -resolver <res> -h[elp] -keep
  -version -describe -p[roperties] <file> -user <user>
  -password <pwd> -role <role> -service <url> -{ssl | iiop}
  -credfile <file> -useservicename -temp <dir> <EJBjarfile>
ejbdescriptor -{parse | dump} <infile> <outfile>
java2rmi_iiop -no_bind -no_comments -no_examples -no_tie -wide
  -root_dir <dir> -verbose -version -W <X>
java2idl
modifyprops -{o[ci8] | t[hin]} -u[ser] <user>/<pwd>@<netserv>
  {<key> <val> | <key> -delete}

```