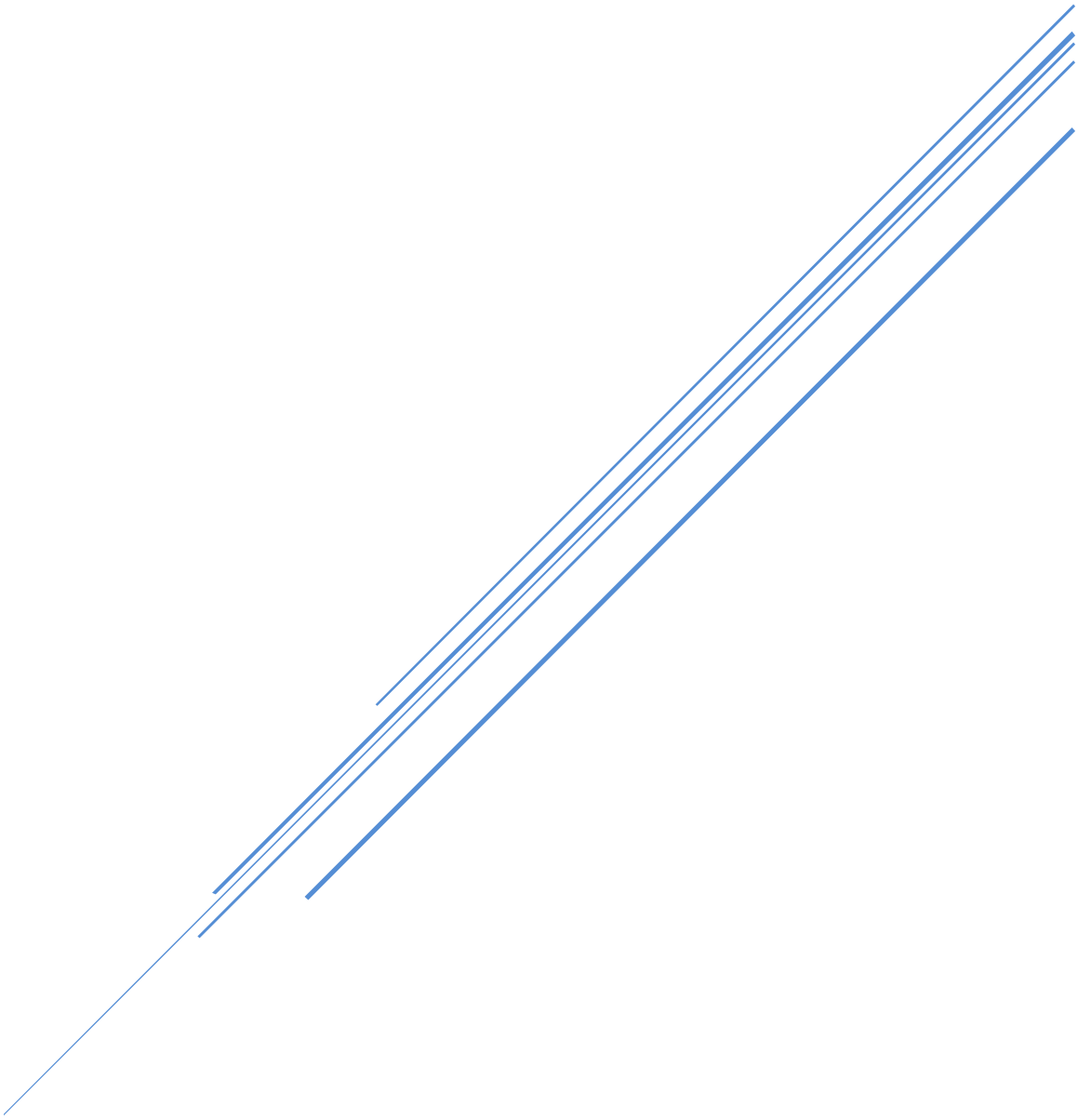
ORACLE DBA QUERIES



TABLES INFORMATION

Data Dictionary Tables and Views

All the table and column information are stored in SYS.TAB$ and SYS.COL$ tables. Oracle has provided data dictionary views to get the information about table and columns

There are three categories of views

USER_%	**This view contain information of the objects owned by the user only** **Example** **USER_TABLES, USER_TAB_COLS**
ALL-%	This view contains information of the objects which the user can access in the database. Example ALL_TABLES, ALL_TAB_COLS
DBA_%	This view contains information of the all objects in the system and these are restricted views which are accessible to the user who have DBA role Example DBA_TABLES, DBA_TAB_COLS

	DBA_% views about table information	ALL_% views about table information	USER_% views about table information
View about column comments	dba_col_comments	all_col_comments	user_col_comments
View about external tables	dba_external_tables	all_external_tables	user_external_tables
View about external tables location	dba_external_locations	all_external_locations	user_external_locations
	dba_partial_drop_tabs	all_partial_drop_tabs	user_partial_drop_tabs
View about table	dba_tables	all_tables	user_tables

information			
View about table column	dba_tab_cols	all_tab_cols	user_tab_cols
	dba_tab_columns	all_tab_columns	user_tab_columns
	dba_tab_col_statistics	all_tab_col_statistics	user_tab_col_statistics
	dba_tab_comments	all_tab_comments	user_tab_comments
	dba_tab_histograms	all_tab_histograms	user_tab_histograms
View about table monitoring	dba_tab_modifications	all_tab_modifications	user_tab_modifications
View table privilege	dba_tab_privs	all_tab_privs	user_tab_privs
	dba_tab_statistics	all_tab_statistics	user_tab_statistics
	dba_tab_stats_history	all_tab_stats_history	user_tab_stats_history
View about unused column in tables	dba_unused_col_tabs	all_unused_col_tabs	dba_unused_col_tabs

To list all tables owned by the current user, type:

select tablespace_name, table_name from user_tables;

To list all tables in a database:

select tablespace_name, table_name from dba_tables;

To list all tables accessible to the current user, type:

select tablespace_name, table_name from all_tables

To describe the table in sqlplus

desc <table_name>

How to determine Table SIZE?

```
select
  owner as "Schema"
  , segment_name as "Object Name"
  , segment_type as "Object Type"
  , round(bytes/1024/1024,2) as "Object Size (Mb)"
  , tablespace_name as "Tablespace"
from dba_segments
where segment_name='<table_name>';
```

INDEX INFORMATION

Data dictionary views on Indexes

DBA_INDEXES **ALL_INDEXES** **USER_INDEXES**	**DBA view describes indexes on all tables in the database. ALL view describes indexes on all tables accessible to the user. USER view is restricted to indexes owned by the user. Some columns in these views contain statistics that are generated by the DBMS_STATS package or ANALYZE statement.**
DBA_IND_COLUMNS **ALL_IND_COLUMNS** **USER_IND_COLUMNS**	These views describe the columns of indexes on tables. Some columns in these views contain statistics that are generated by the DBMS_STATS package or ANALYZE statement.
DBA_IND_EXPRESSIONS **ALL_IND_EXPRESSIONS** **USER_IND_EXPRESSIONS**	These views describe the expressions of function-based indexes on tables.
DBA_IND_STATISTICS **ALL_IND_STATISTICS** **USER_IND_STATISTICS**	These views contain optimizer statistics for indexes.

How to determine the indexes on the table?

```
set page size 50000 verify off echo off

col table_name head 'Table Name' format a20
col index_name head 'Index Name' format a25
col column_name head 'Column Name' format a30
```

```
break on table_name on index_name

select table_name, index_name, column_name
from   all_ind_columns
where table_name like upper('&Table_Name')
order by table_name, index_name, column_position
/
```

How to determine index SIZE?

Size of INDEX

```
select segment_name,sum(bytes)/1024/1024/1024 as "SIZE in GB" from
user_segments where segment_name='INDEX_NAME' group by segment_name;
OR
select owner,segment_name,sum(bytes)/1024/1024/1024 as "SIZE in GB" from
dba_segments where owner='SCHEMA_NAME' and
segment_name='INDEX_NAME' group by owner,segment_name;
```

List of Size of all INDEXES of a USER

```
select segment_name,sum(bytes)/1024/1024/1024 as "SIZE in GB" from
user_segments where segment_type='INDEX' group by segment_name order by
"SIZE in GB" desc;
 OR
select owner,segment_name,sum(bytes)/1024/1024/1024 as "SIZE in GB" from
dba_segments where owner='SCHEMA_NAME' and segment_type='INDEX' group
by owner,segment_name order by "SIZE in GB" desc;
```

Sum of sizes of all indexes

```
select owner,sum(bytes)/1024/1024/1024 as "SIZE in GB" from dba_segments
where owner='SCHEMA_NAME' and segment_type='INDEX' group by owner;
```

VIEW INFORMATION

Dictionary Views for seeing the View data

View details can be queried from the dictionary by querying either USER_VIEWS, ALL_VIEWS or DBA_VIEWS. Views are useful for security and information hiding,

but can cause problems if nested too deeply. Some of the advantages of using views:

- Reduce the complexity of SQL statements
- Share only specific rows in a table with other users
- Hide the NAME and OWNER of the base table
  - There are three categories of views

USER_%	**This view contain information of the objects owned by the user only** **Example** **USER_TABLES,USER_TAB_COLS**
ALL-%	This view contains information of the objects which the user can access in the database. Example ALL_TABLES,ALL_TAB_COLS
DBA_%	This view contain information of the all objects in the system and these are restricted views which are accessible to the user who have DBA role Example DBA_TABLES,DBA_TAB_COLS

	DBA_% views about table information	ALL_% views about table information	USER_% views about table information
Column which can be updated	DBA_UPDATABLE_COLUMNS	ALL_UPDATABLE_COLUMNS	USER_UPDATABLE_COLUMNS
View about view information	dba_views	all_views	user_views

To list all views owned by the current use

select view_name from user_views;

To list all views in a database:

Select owner,view_name from dba_views;

To list views accessible to the current user:

select view_name from all_views

To describe the view in sqlplus

desc <view_name>

How to determine the query of the already created view

Query the TEXT column of table DBA_VIEWS.

Syntax:
SQL> set long 10000

```
SQL> select TEXT
  2  FROM DBA_VIEWS
  3  where OWNER = '<owner_name>'
  4  and VIEW_NAME  = '<view_name>';
```

How to extract the view definition (DDL statements) from an Oracle database without having to go through a stack of dictionary views

Syntax:
SQL> set long 1000
SQL> set pagesize 0

select DBMS_METADATA.GET_DDL('VIEW','<view_name>') from DUAL;

SEQUENCE INFORMATION

Dictionary Views for seeing the sequence data

sequence details can be queried from the dictionary by querying either USER_SEQUENCES, ALL_ SEQUENCES or DBA_ SEQUENCES. There are three categories of views

USER_%	**This view contain information of the objects owned by the user only** **Example** **USER_TABLES,USER_TAB_COLS**

ALL-%	This view contains information of the objects which the user can access in the database. Example ALL_TABLES,ALL_TAB_COLS
DBA_%	This view contain information of the all objects in the system and these are restricted views which are accessible to the user who have DBA role Example DBA_TABLES,DBA_TAB_COLS

	DBA_% views about sequences information	**ALL_% views about sequences information**	**USER_% views about sequences information**
View about sequences information	dba_sequences	all_ sequences	user_ sequences

To list all sequences owned by the current use

select sequence_name from user_sequences;

To list all sequences in a database:

Select owner, sequence_name from dba_sequences;

To list sequences accessible to the current user:

select sequence_name from all_sequences

How to determine the all information about the sequence?

```
select sequence_name,min_value,max_value,increment_by,last_number
  FROM DBA_SEQUENCES
  where OWNER = '<owner_name>'
  and sequence_NAME  = '<sequence_name>';
```

The last_number column display the next available sequence number if no cache is specified

How to extract the sequence definition (DDL statements) from an Oracle

database without having to go through a stack of dictionary views

Syntax:
SQL> set long 1000
SQL> set pagesize 0

select DBMS_METADATA.GET_DDL('SEQUENCE','<sequence_name>') from DUAL;

Impact of caching the sequences

Sequence are cached with the purpose of improving the fetch performance. In RAC,each instance stores the cache values

We can have gaps in sequence when using cache due to following reasons

a) Rollback occurs
b) System crash or instance crash
c) Sequence is used in another table

PRIVILEGES INFORMATION

Data Dictionary Tables and Views

Oracle has provided data dictionary views to get the information about privileges

There are three categories of views

USER_%	**This view contain information of the objects owned by the user only** **Example** **USER_TABLES, USER_TAB_COLS**
ALL-%	This view contains information of the objects which the user can access in the database. Example ALL_TABLES,ALL_TAB_COLS
DBA_%	This view contains information of the all objects in the system and these are restricted views which are accessible to the user who have DBA role Example DBA_TABLES,DBA_TAB_COLS

Checking Privileges Views

USER_ROLE_PRIVS	**Roles accessible by the user**
ROLE_SYS_PRIVS	System privilege's granted to Role
ROLE_TAB_PRIVS	Table privilege's granted to Role
USER_TAB_PRIVS_MADE	Objects privileges granted on the user's objects
USER_TAB_PRIVS_RECD	Objects privileges granted to the user
USER_COL_PRIVS_MADE	Objects privileges granted on the columns of the user's objects
USER_COL_PRIVS_RECD	Objects privileges granted to the user on the specific column's
USER_SYS_PRIVS	Lists system privileges granted to the user
ALL_OBJECTS	displays all of the objects to which the user has access to
ALL_COL_PRIVS_MADE	displays all of the grants on columns that the user owns or that the user has granted
USER_OBJECTS	displays only the objects owned by the user.
USER_TABLES	displays only the tables owned by the user.
USER_VIEWS	displays only the views owned by the user.
DICTIONARY	this view provides descriptions of the data dictionary tables and views that are accessible to the user
TABLE_PRIVILEGES	**Displays the grants on objects**: ○ When role or a PUBLIC is grantee ○ The user has granted. ○ That have been granted to the user. ○ That the user owns.

Determine Roles and System/Table Privileges Granted to Users

```
set lines 110 pages 1000 ver off
col role for a16
col pv for a75 hea 'PRIVILEGE OR ROLE'
bre on role on type skip 1
define usercheck = 'SH'
select grantee, 'ROL' type, granted_role pv
from dba_role_privs where grantee = '&usercheck' union
select grantee, 'PRV' type, privilege pv
from dba_sys_privs where grantee = '&usercheck' union
select grantee, 'OBJ' type,
max(decode(privilege,'WRITE','WRITE,'))||max(decode(privilege,'READ','READ'))||
max(decode(privilege,'EXECUTE','EXECUTE'))||max(decode
(privilege,'SELECT','SELECT'))||
max(decode(privilege,'DELETE',',DELETE'))||max(decode
(privilege,'UPDATE',',UPDATE'))||
max(decode(privilege,'INSERT',',INSERT'))||' ON '||object_type||'
"'||a.owner||'.'||table_name||'"' pv
from dba_tab_privs a, dba_objects b
where a.owner=b.owner and a.table_name = b.object_name and
a.grantee='&usercheck'
group by a.owner,table_name,object_type,grantee union
select username grantee, '---' type, 'empty user ---' pv from dba_users
where not username in (select distinct grantee from dba_role_privs) and
not username in (select distinct grantee from dba_sys_privs) and
not username in (select distinct grantee from dba_tab_privs) and username like
'%&usercheck%'
group by username
order by grantee, type, pv;
```

Determine the system privs given to the user

```
SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS
  WHERE GRANTEE = 'USER';
```

Checking which table privileges are granted by you to other users.

```
SELECT * FROM USER_TAB_PRIVS_MADE
```

Checking which table privileges are granted to you by other users

```
SELECT * FROM USER_TAB_PRIVS_RECD;
```

Checking which column level privileges are granted by you to other users.

```
SELECT * FROM USER_COL_PRIVS_MADE
```

Checking which column level privileges are granted to you by other users

SELECT * FROM USER_COL_PRIVS_RECD;

Checking which privileges are granted to roles

SELECT * FROM USER_ROLE_PRIVS;

TABLESPACE INFORMATION

Dictionary views for Viewing Tablespace Information

View	Description
V$TABLESPACE	Name and number of all tablespaces from the control file.
DBA_TABLESPACES, USER_TABLESPACES	Descriptions of all (or user accessible) tablespaces.
DBA_SEGMENTS, USER_SEGMENTS	Information about segments within all (or user accessible) tablespaces.
DBA_EXTENTS, USER_EXTENTS	Information about data extents within all (or user accessible) tablespaces.
DBA_FREE_SPACE, USER_FREE_SPACE	Information about free extents within all (or user accessible) tablespaces.
V$DATAFILE	Information about all datafiles, including tablespace number of owning tablespace.
V$TEMPFILE	Information about all tempfiles, including tablespace number of owning tablespace.
DBA_DATA_FILES	Shows files (datafiles) belonging to tablespaces.
DBA_TEMP_FILES	Shows files (tempfiles) belonging to temporary tablespaces.
V$TEMP_EXTENT_MAP	Information for all extents in all locally managed temporary tablespaces.
V$TEMP_EXTENT_POOL	For locally managed temporary tablespaces: the state of temporary space cached and used for by each instance.
V$TEMP_SPACE_HEADER	Shows space used/free for each tempfile.
DBA_USERS	Default and temporary tablespaces for all users.
DBA_TS_QUOTAS	Lists tablespace quotas for all users.

V$SORT_SEGMENT	Information about every sort segment in a given instance. The view is only updated when the tablespace is of the TEMPORARY type.
V$SORT_USER	Temporary sort space usage by user and temporary/permanent tablespace.

To list Tablespaces and all important Properties:

To list the names and various other of all tablespaces in a database, use the following query on the DBA_TABLESPACES view:

```
SELECT TABLESPACE_NAME "TABLESPACE",
   EXTENT_MANAGEMENT, FORCE_LOGGING, BLOCK_SIZE,
SEGMENT_SPACE_MANAGEMNENT
   FROM DBA_TABLESPACES;
```

To list the Datafiles and Associated Tablespaces of a Database

To list the names, sizes, and associated tablespaces of a database, enter the following query on the DBA_DATA_FILES view

```
SELECT FILE_NAME, BLOCKS, TABLESPACE_NAME
   FROM DBA_DATA_FILES;
```

To display Statistics for Free Space (Extents) of Each Tablespace

To produce statistics about free extents and coalescing activity for each tablespace in the database, enter the following query:

```
SELECT TABLESPACE_NAME "TABLESPACE", FILE_ID,
   COUNT(*)    "PIECES",
   MAX(blocks) "MAXIMUM",
   MIN(blocks) "MINIMUM",
   AVG(blocks) "AVERAGE",
   SUM(blocks) "TOTAL"
   FROM DBA_FREE_SPACE
GROUP BY TABLESPACE_NAME, FILE_ID;
```

How to check highest allocated extent?

```
column file_name format a50;
column tablespace_name format a15;
column highwater format 9999999999;
set pagesize 9999

select a.tablespace_name
,a.file_name
,(b.maximum+c.blocks-1)*d.db_block_size highwater
from dba_data_files a
,(select file_id,max(block_id) maximum
from dba_extents
group by file_id) b
,dba_extents c
,(select value db_block_size
from v$parameter
where name='db_block_size') d
where a.file_id = b.file_id
and c.file_id = b.file_id
and c.block_id = b.maximum
order by a.tablespace_name,a.file_name
/
```

To check the free SPACE, largest free chunck and no of free chunck in tablespace.

```
set feedback off
set echo off
set numwidth 15
set linesize 150
set pages 1000

Accept tname Prompt "Enter Tablespace Name : "
Select (Sum(bytes)/1024/1024) Free_space_MB,(max(bytes)/1024/1024)
Largest_Free_chunck_MB,count(*) No_of_free_chunk
from dba_free_space where tablespace_name=upper('&tname');
```

To check the total space allocated to tablespace.

```
Select (sum(bytes)/1024/1024) Space_allocated
from dba_data_files
where tablespace_name=upper('&tname');
```

To check all tablespace information in the database

```
set echo off feedback off verify off pages 60

col tablespace_name format a16 head 'Tablespace Name'
col initial_extent format 99,999,999 head 'Initial|Extent(K)'
col next_extent format 99,999,999 head 'Next|Extent(K)'
--col min_extents format 999 head 'Min|Ext'
col max_extents format a4 head 'Max|Ext'
col pct_increase format 999 head 'Pct|Inc'
col extent_management format a10 head 'Extent|Management'
col allocation_type format a10 head 'Allocation|Type'
col status format a7 head 'Status'

select tbs.tablespace_name
,       tbs.initial_extent
,       tbs.next_extent
--,     tbs.min_extents
,       decode(tbs.max_extents,2147483645,'UL',tbs.max_extents) max_extents
,       tbs.pct_increase
,       tbs.extent_management
,       tbs.allocation_type
,       tbs.status
from   dba_tablespaces tbs
order by 1
/
```

VIEWS AND TABLE TO VIEW ENQUEUE AND LOCKS

a) V$session and v$session_wait

When is session is waiting on enqueue or lock, this can be session from V$session (in 11g and above) and v$session_wait

We can use below query to obtain all the enqueue in the system

```
Select * from v$session_wait where event like ‘enq%’;

The parameter of the enqueue wait event  has following meaning

P1: resource type and mode wanted

P2:ID1 of the resource

P3: ID2 of the resource
```

```
Select event,p1, p2,p3 from v$session_wait  where wait_time=0 and event like 'enq%';
```

b) V$lock is another useful view to check enqueue 's

i) V$lock list all the lock structure currently held in the system

ii) The column type ,id1 and id2 represent the resource type ,id1 and id2 of the resource structure.so it can be joined with V$resource which contains the list of all the resource structure

iii) LMODE and request tells us which queue (owner,converter,waiters) is the session

LMODE	Request	Queue name
> 0	=0	Owner
=0	> 0	Waiter
> 0	> 0	Converter

Below query can be used to find holder and waiter

```
SELECT inst_id,DECODE(request,0,'Holder: ','Waiter: ')||sid sess,
           id1, id2, lmode, request, type
       FROM V$LOCK
      WHERE (id1, id2, type) IN
              (SELECT id1, id2, type FROM V$LOCK WHERE request>0)
      ORDER BY id1, request
     ;
```

In case of RAC

```
SELECT inst_id,DECODE(request,0,'Holder: ','Waiter: ')||sid sess,
           id1, id2, lmode, request, type
       FROM GV$LOCK
      WHERE (id1, id2, type) IN
              (SELECT id1, id2, type FROM gV$LOCK WHERE request>0)
      ORDER BY id1, request
   ;
```

c) V$locked_object is another useful view

It contains all the TM locks in the database. It gives the transaction slot, OS process is and session id of the session which is holding the TM locks

d) There are several views which can be used to find the locks information. These views are created by catblock.sql

DBA_LOCKS	**Show all the locks like v$lock**
DBA_DML_LOCKS	Shows all DML™ locks held or being requested
DBA_DDL_LOCKS	Shows all DDL locks held or being requested
DBA_WAITERS	Shows all sessions waiting on, but not holding waited for locks
DBA_BLOCKERS	Shows non-waiting sessions holding locks being waited-on

Query to find out waiting session and holding sessions

```
set linesize 1000
column waiting_session heading 'WAITING|SESSION'
column holding_session heading 'HOLDING|SESSION'
column lock_type format a15
column mode_held format a15
column mode_requested format a15

select
   waiting_session,
   holding_session,
   lock_type,
   mode_held,
   mode_requested,
```

```
   lock_id1,
   lock_id2
from
   dba_waiters
/
```

Query to find out all the locked objects

```
set term on;
set lines 130;
column sid_ser format a12 heading 'session,|serial#';
column username format a12 heading 'os user/|db user';
column process format a9 heading 'os|process';
column spid format a7 heading 'trace|number';
column owner_object format a35 heading 'owner.object';
column locked_mode format a13 heading 'locked|mode';
column status format a8 heading 'status';
select
    substr(to_char(l.session_id)||','||to_char(s.serial#),1,12) sid_ser,
    substr(l.os_user_name||'/'||l.oracle_username,1,12) username,
    l.process,
    p.spid,
    substr(o.owner||'.'||o.object_name,1,35) owner_object,
    decode(l.locked_mode,
            1,'No Lock',
            2,'Row Share',
            3,'Row Exclusive',
            4,'Share',
            5,'Share Row Excl',
            6,'Exclusive',null) locked_mode,
    substr(s.status,1,8) status
from
    v$locked_object l,
    all_objects     o,
    v$session       s,
    v$process       p
where
    l.object_id = o.object_id
and l.session_id = s.sid
and s.paddr      = p.addr
and s.status != 'KILLED'
/
```

Query to find the blocking session for Library Cache lock

```
select /*+ all_rows */ w1.sid waiting_session,
```

```
h1.sid holding_session,
w.kgllktype lock_or_pin,
w.kgllkhdl address,
decode(h.kgllkmod, 0, 'None', 1, 'Null', 2, 'Share', 3, 'Exclusive',
'Unknown') mode_held,
decode(w.kgllkreq, 0, 'None', 1, 'Null', 2, 'Share', 3, 'Exclusive',
'Unknown') mode_requested
from dba_kgllock w, dba_kgllock h, v$session w1, v$session h1
where
(((h.kgllkmod != 0) and (h.kgllkmod != 1)
and ((h.kgllkreq = 0) or (h.kgllkreq = 1)))
and
(((w.kgllkmod = 0) or (w.kgllkmod= 1))
and ((w.kgllkreq != 0) and (w.kgllkreq != 1))))
and w.kgllktype = h.kgllktype
and w.kgllkhdl = h.kgllkhdl
and w.kgllkuse = w1.saddr
and h.kgllkuse = h1.saddr
/
```

To find the holder of the CF enqueue, the following query can be used :

```
select l.sid, p.program, p.pid, p.spid, s.username, s.terminal, s.module, s.action,
s.event, s.wait_time, s.seconds_in_wait, s.state
from v$lock l, v$session s, v$process p
where l.sid = s.sid
and s.paddr = p.addr
and l.type='CF'
and l.lmode >= 5;
```

To find the session waiting to get the CF enqueue, the following query can be used :

```
select l.sid, p.program, p.pid, p.spid, s.username, s.terminal, s.module, s.action,
s.event, s.wait_time, s.seconds_in_wait, s.state
from v$lock l, v$session s, v$process p
where l.sid = s.sid
and s.paddr = p.addr
and l.type='CF'
and l.request >= 5;
```

SESSION QUERIES

How to kill multiple user session using some condition?

select 'alter system kill session '||''''||sid||','||serial#||''''';' from v$session

where <condition>;

Example

Kill all the session in database except of oracle user

select 'alter system kill session '||''''||sid||','||serial#||''''';' from v$session

where upper(substr(osuser,1,8)) not in (‘oracle’)

How to find the active session in the database of particular user?

```
SELECT substr(SID,1,6) sid,SERIAL#,substr(OSUSER,1,30)
OSUSER,MACHINE,STATUS,PROGRAM
 FROM V$SESSION
WHERE USERNAME = (‘&1’) and status='ACTIVE'
/
```

How to report the count of session in Database?

```
SET LINESIZE 85
SET PAGESIZE 10000
SET FEEDBACK OFF
COLUMN "ACTIVE" FORMAT 999999 HEADING "ACTIVE"
COLUMN "INACTIVE" FORMAT 999999 HEADING "INACTIVE"
COLUMN "CACHED" FORMAT 999999 HEADING "CACHED"
COLUMN "KILLED" FORMAT 999999 HEADING "KILLED"
COLUMN "SNIPED" FORMAT 999999 HEADING "SNIPED"
BREAK ON Report SKIP 1
COMPUTE SUM OF ACTIVE ON Report
COMPUTE SUM OF INACTIVE ON Report
COMPUTE SUM OF CACHED ON Report
COMPUTE SUM OF KILLED ON Report
COMPUTE SUM OF SNIPED ON Report
SELECT NVL(USERNAME, 'BACKGROUND PROCESS') "USER NAME",
COUNT(DECODE(STATUS,'ACTIVE','1')) "ACTIVE",
COUNT(DECODE(STATUS,'INACTIVE','1')) "INACTIVE",
COUNT(DECODE(STATUS,'CACHED','1')) "CACHED",
COUNT(DECODE(STATUS,'KILLED','1')) "KILLED",
COUNT(DECODE(STATUS,'SNIPED','1')) "SNIPED"
FROM V$SESSION
GROUP BY USERNAME
/
```

BACKUP QUERIES

How to find the datafile in Hotbackup mode?

```
column file# format 99999999
column name format a50
select d.file#,d.name,b.status
from v$DATAFILE d,v$BACKUP b
where d.file#=b.file# and b.status='ACTIVE'
/
```

This SQL script creates sql files for begin and end tablespace backup

```
cat ts_back.sql
set pagesize 0 feed off echo off termout off
spool /home/oracle/create/begin_backup.sql
select 'set echo on verify on feed on termout on trimspool on' from dual
/
select 'column NAME format A31' from dual
/
select 'alter system switch logfile;' from dual
/
select 'select SEQUENCE#,ARCHIVED,STATUS from v$log where STATUS =
''ACTIVE'';' from dual
/
select 'alter tablespace '||tablespace_name||' begin backup;' from
dba_tablespaces
/
select 'exit' from dual
/
spool off
spool /home/oracle/create/end_backup.sql
select 'set echo on verify on feed on termout on' from dual
/
select 'alter tablespace '||tablespace_name||' end backup;' from dba_tablespaces
/
select 'alter system switch logfile;' from dual
/
select 'select NAME,RECID from v$archived_log where RECID=(select
SEQUENCE#-1 from v$log  where STATUS = ''ACTIVE'');' from dual
/
select 'exit' from dual
/
spool off
set termout on lines 180
```

```
select 'Output files are in /home/oracle/create/begin_backup.sql and
/home/oracle/create/end_backup.sql' from dual
/
select ' ' from dual
/
exit
/
```

NORMAL PERFORMANCE AND MAINTENANCE QUERIES

Average Buffer Hit ratio from the time of start of database

```
select (
          (1- (sum(decode(name,'physical reads',value,0))/
              (sum(decode(name,'db block gets',value,0))
             + (sum(decode(name,'consistent gets',value,0))
         ))))*100) "Buffer Hit Ratio"
from v$sysstat
/
```

How to find Chained rows count?

```
col "Percent Chained" format 99.99
select   OWNER,
        TABLE_NAME,
        nvl(CHAIN_CNT,0) "Chained Rows",
        nvl(NUM_ROWS,0) "Total Rows",
        (CHAIN_CNT/NUM_ROWS)*100 "Percent Chained"
from     dba_tables
where    owner not in ('SYS','SYSTEM')
and      nvl(CHAIN_CNT,0) > 0
order    by (CHAIN_CNT/NUM_ROWS) desc
/
```

How the file datafile information tablespace wise

```
select file_name,tablespace_name,sum(bytes)/1024/1024 from dba_data_files
group by file_name,tablespace_name order by tablespace_name,file_name
/
```

How to derive the transaction information per sid

```
select s.sid,s.username,s.osuser,t.UBABLK,t.USED_UBLK,
```

```
s.terminal,s.status,t.start_time,
s.type,s.program
from v$session s,v$transaction t
where s.taddr=t.addr
order by t.start_time
/
```

How to find object modified in last 1 day

```
select   OWNER,
        OBJECT_NAME,
        OBJECT_TYPE,
        to_char(LAST_DDL_TIME,'MM/DD/YYYY HH24:MI:SS'),
        to_char(CREATED,'MM/DD/YYYY HH24:MI:SS'),
        STATUS
from     dba_objects
where    (SYSDATE - LAST_DDL_TIME) < 1
order by LAST_DDL_TIME DESC;
```

How to find top 10 longest idle inactive session

```
col osuser format a10 trunc
col LastCallET     format 99,999
col sid format 9999
col username format a10 trunc
col uprogram format a25 trunc
col umachine format a10 trunc
set linesize 132
set verify off
select * from (
select to_char(s.logon_time, 'mm/dd hh:mi:ssAM') loggedon,
  s.sid, s.status,
  floor(last_call_et/60) "LastCallET",
 s.username, s.osuser,
 p.spid, s.module || ' - ' || s.program uprogram,
s.machine, s.sql_hash_value
from v$session s, v$process p
where p.addr = s.paddr
  and s.type = 'USER'
  and s.username is not null
  and s.status = 'INACTIVE'
order by 4 desc)
where rownum < 11
```

How to find the sqlplan for the sql

```
set linesize 9999
column QUERY format a999
set pages 250
set head off
```

```
set verify off
select id,lpad(' ',2*(depth-1)) || depth ||'.'  || nvl(position,0) || ' '|| operation || '
'|| options || ' '|| object_name ||' '
||'cost=  '|| to_char(cost)||' '|| optimizer "QUERY"
  from v$sql_plan
 where hash_value = &1
 order by child_number,id
/
```

How to Check whether stats is current for the sql

```
set lin 1000
set verify off
col owner format a15
col object_name format a25
col object_type format a12
col "LAST ANALYZED" format a13

select do.OWNER,do.OBJECT_NAME,OBJECT_TYPE,
decode (OBJECT_TYPE,'TABLE'  , (Select LAST_ANALYZED from dba_tables where
owner=do.owner and TABLE_NAME=do.object_name)  ,
                'INDEX'  , (Select LAST_ANALYZED from dba_indexes where
owner=do.owner and INDEX_NAME=do.object_name) ,
                'UNKNOWN') "LAST ANALYZED",STATUS
from   DBA_OBJECTS do
where  OBJECT_TYPE in ('TABLE','INDEX')
and    (OWNER,OBJECT_NAME) in (select OBJECT_OWNER,OBJECT_NAME from
V$SQL_PLAN where HASH_VALUE=&1)
/
```

How to check Stats of Table

```
col num_rows format 999999990 heading 'ROWS'
col chain_cnt format 99990 heading 'CHAIN|COUNT'
col avg_row_len format 9990 heading 'AVG|ROW|SIZE'
col blocks format 9999990 heading 'USED|BLOCKS'
col empty_blocks format 999990 heading 'EMPTY|BLOCKS'
col avg_space format 9990 heading 'AVG|FREE|SPACE'
set verify off
rem
break on report on owner skip 1
compute sum of num_rows blocks empty_blocks on report owner
rem
select owner, table_name, num_rows, chain_cnt, avg_row_len,
       blocks, empty_blocks, avg_space,last_analyzed
  from sys.dba_tables
```

```
 where (owner,table_name) in (select OBJECT_OWNER,OBJECT_NAME from
V$SQL_PLAN where HASH_VALUE= &&1)
 order by owner, table_name
/
```

How to check Stats of Index

```
rem
set linesize 200
set pages 250
set verify off
col blevel format 99
col table_name format a22 heading 'TABLE NAME'
col u format a1 heading 'U'
col index_name format a25 heading 'INDEX NAME'
col column_name format a23 heading 'COLUMN NAME'
col column_position format 99 heading 'SEQ'
col column_length format 9999 heading 'LEN'
col leaf_blocks format 999990 heading 'LEAF|BLOCKS'
col distinct_keys format 9999990 heading 'DISTINCT|KEYS'
col avg_leaf_blocks_per_key format 999990 heading 'LEAF|BLKS|/KEY'
col avg_data_blocks_per_key format 999990 heading 'DATA|BLKS|/KEY'
rem
break on table_name skip 1 on index_name on u
rem
select i.table_name,i.blevel, i.leaf_blocks,
i.distinct_keys,i.avg_leaf_blocks_per_key, i.avg_data_blocks_per_key,
       decode( i.uniqueness, 'NONUNIQUE', null, 'UNIQUE', 'U', 'BITMAP', 'B', '?' ) u,
       i.index_name,i.last_analyzed, i.CLUSTERING_FACTOR
  from sys.dba_ind_columns c, sys.dba_indexes i
 where (i.table_owner,i.table_name) in (select OBJECT_OWNER,OBJECT_NAME
from V$SQL_PLAN where HASH_VALUE= &&1)
   and i.owner = c.index_owner
   and i.index_name = c.index_name
 order by i.table_owner, i.table_name, i.index_name, c.column_position
/
```

How to check Sql statistics

```
select EXECUTIONS,DISK_READS,BUFFER_GETS,
CPU_TIME,ELAPSED_TIME,ROWS_PROCESSED,INVALIDATIONS,PARSE_CALLS
from v$sql
```

```
where HASH_VALUE= &1
/
```

How to find details information about sql From Memory

```
set pages 1000 lines 200
col first_load_time for a20
col last_load_time for a20
col outline_category for a20
col sql_profile for a32
select sql_id, child_number, plan_hash_value, first_load_time, last_load_time,
outline_category, sql_profile, executions,
trunc(decode(executions, 0, 0, rows_processed/executions)) rows_avg,
 trunc(decode(executions, 0, 0, fetches/executions)) fetches_avg,
trunc(decode(executions, 0, 0, disk_reads/executions)) disk_reads_avg,
 trunc(decode(executions, 0, 0, buffer_gets/executions)) buffer_gets_avg,
trunc(decode(executions, 0, 0, cpu_time/executions)) cpu_time_avg,
 trunc(decode(executions, 0, 0, elapsed_time/executions)) elapsed_time_avg,
 trunc(decode(executions, 0, 0, application_wait_time/executions))
apwait_time_avg,
 trunc(decode(executions, 0, 0, concurrency_wait_time/executions))
cwait_time_avg,
trunc(decode(executions, 0, 0, cluster_wait_time/executions)) clwait_time_avg,
trunc(decode(executions, 0, 0, user_io_wait_time/executions)) iowait_time_avg,
 trunc(decode(executions, 0, 0, plsql_exec_time/executions)) plsexec_time_avg,
 trunc(decode(executions, 0, 0, java_exec_time/executions)) javexec_time_avg
from v$sql
 where sql_id = '&sql_id'
 order by sql_id, child_number;
```

How to find details information about sql From AWR

```
set pages 1000 lines 200
col sql_profile for a32
 select sql_id, snap_id, plan_hash_value, sql_profile, executions_total,
trunc(decode(executions_total, 0, 0, rows_processed_total/executions_total))
rows_avg,
 trunc(decode(executions_total, 0, 0, fetches_total/executions_total)) fetches_avg,
 trunc(decode(executions_total, 0, 0, disk_reads_total/executions_total))
disk_reads_avg,
 trunc(decode(executions_total, 0, 0, buffer_gets_total/executions_total))
buffer_gets_avg,
```

```
 trunc(decode(executions_total, 0, 0, cpu_time_total/executions_total))
cpu_time_avg,
 trunc(decode(executions_total, 0, 0, elapsed_time_total/executions_total))
elapsed_time_avg,
 trunc(decode(executions_total, 0, 0, iowait_total/executions_total))
iowait_time_avg,
 trunc(decode(executions_total, 0, 0, clwait_total/executions_total))
clwait_time_avg,
 trunc(decode(executions_total, 0, 0, apwait_total/executions_total))
apwait_time_avg,
 trunc(decode(executions_total, 0, 0, ccwait_total/executions_total))
ccwait_time_avg,
 trunc(decode(executions_total, 0, 0, plsexec_time_total/executions_total))
plsexec_time_avg,
 trunc(decode(executions_total, 0, 0, javexec_time_total/executions_total))
javexec_time_avg
from dba_hist_sqlstat
 where sql_id = '&sql_id'
 order by sql_id, snap_id;
```