**DB Monitoring & Performance Script**

The Monitoring of predefined events that generates a message or warning when a certain threshold has been exceeded. This is done in an effort to ensure that an issue doesn't become a problem. The database monitoring is required for the following reason:

–        Smooth running of production

–        Keeping an eye on development

–        Database performance

–        In Support of an SLA (service level agreement)

**Types of DB Monitoring**

1. Status
2. Performance
3. Trend Analysis

**Status Monitoring:**

Monitor the current status of an event and reports when it exceeds a defined threshold.

**Database:**

–        Database/Listener

–        Monitor Alert. log Message on regular basis.

–        Check all last night backup is successful.

–        Tablespace/Datafiles full or Fragmented.

–        Identify bad growth of segment.

–        Identify at least 1 top resource consuming query

–        Monitor Locking

–        Check Maximum Extent about to be reached.

–        Redo log Tracking

–        UNDO and Temp Segment Free space.

–        Monitor Running Job

–        Tracking DB User/Session Information.

–        Important Object Information

**OS:**

–        SGA/PGA information

–        CPU Usage Information

–        Memory Utilization

–        Disk Utilization

**Performance Monitoring:**

Monitor a defined set of performance statistics. This is done in an effort to maintain the best possible DB performance.

**Trend Analysis Monitoring:**

Collect the historical data for specified events and analyze these data on schedule basis to reveal any potential problems. For Example watching growth of data in a tablespace and predicting when it will fill.

Apart from the above checklist some of the other checklist a DBA are using. It is depend on the requirement. I am mentioning here some of the related query and scripts. It is fully related to DB Monitoring Purpose.  
**Note:** *Keep every one informed specially your senior or Junior DBA, System Admin, Manager and do not forget to document very important update.*

**Database Information:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Track OS Reboot Time:**  
net statistics server  
systeminfo | find "Up Time"  -- to find system last uptime  
systeminfo | find "System Boot Time"  -- to find system boot time  
net statistics workstation | find "Statistics" Workstation Statistics for \\A5541TAG-WKS   --perticular workstation statistics  
**Database and Instance Last start time:**  
SELECT to\_char(startup\_time,'DD-MON-YYYY HH24:MI:SS') "DB Startup Time"  
FROM   sys.v\_$instance;  
SELECT SYSDATE-logon\_time "Days", (SYSDATE-logon\_time)\*24 "Hours"  
from  sys.v\_$session where  sid=1;  
**Track Database Version:**  
SELECT \* from v$version;  
**Track Database Name and ID information:**  
SELECT DBID, NAME FROM V$DATABASE;‎  
**Track Database Global Name information:**  
SELECT \* FROM GLOBAL\_NAME;‎  
**Track Database Instance name:**  
SELECT INSTANCE\_NAME FROM V$INSTANCE;‎  
**Track Database Host Details:**  
SELECT UTL\_INADDR.GET\_HOST\_ADDRESS, UTL\_INADDR.GET\_HOST\_NAME FROM DUAL;  
**Display information about database services**  
SELECT name,  network\_name FROM   dba\_services ORDER BY name;

**Track Database Present Status:**

SELECT created, RESETLOGS\_TIME, Log\_mode FROM V$DATABASE;  
**DB Character Set Information:**  
Select \* from nls\_database\_parameters;  
**Track Database default information:**  
Select username, profile, default\_tablespace, temporary\_tablespace from dba\_users;  
**Track Total Size of Database:**  
select a.data\_size+b.temp\_size+c.redo\_size "Total\_Size (GB)"  
from ( select sum(bytes/1024/1024/1024) data\_size  
         from dba\_data\_files ) a, ( select nvl(sum(bytes/1024/1024/1024),0) temp\_size  
         from dba\_temp\_files ) b, ( select sum(bytes/1024/1024/1024) redo\_size  
         from sys.v\_$log ) c;  
**Total Size of Database with free space:**  
Select round(sum(used.bytes) / 1024 / 1024/1024 ) || ' GB' "Database Size",round(free.p / 1024 / 1024/1024) || ' GB' "Free space"  
from (select bytes from v$datafile  
      union all  
      select bytes from v$tempfile  
      union all  
      select bytes from v$log) used, (select sum(bytes) as p from dba\_free\_space) free group by free.p;  
**Track Database Structure:**  
select name from   sys.v\_$controlfile;  
select group#,member from   sys.v\_$logfile;  
Select F.file\_id Id, F.file\_name name, F.bytes/(1024\*1024) Mbyte,  
decode(F.status,'AVAILABLE','OK',F.status) status, F.tablespace\_name Tspace  
from   sys.dba\_data\_files F  
order by tablespace\_name;  
**Tablespace/Datafile/Temp/UNDO Information:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Track Tablespace Used/Free Space:**  
SELECT /\* + RULE \*/  df.tablespace\_name "Tablespace",  df.bytes / (1024 \* 1024) "Size (MB)",  
       SUM(fs.bytes) / (1024 \* 1024) "Free (MB)", Nvl(Round(SUM(fs.bytes) \* 100 / df.bytes),1) "% Free", Round((df.bytes - SUM(fs.bytes)) \* 100 / df.bytes) "% Used"  
  FROM dba\_free\_space fs, (SELECT tablespace\_name,SUM(bytes) bytes  
          FROM dba\_data\_files  
         GROUP BY tablespace\_name) df  
 WHERE fs.tablespace\_name (+)  = df.tablespace\_name  
 GROUP BY df.tablespace\_name,df.bytes  
UNION ALL  
SELECT /\* + RULE \*/ df.tablespace\_name tspace,  
       fs.bytes / (1024 \* 1024), SUM(df.bytes\_free) / (1024 \* 1024), Nvl(Round((SUM(fs.bytes) - df.bytes\_used) \* 100 / fs.bytes), 1), Round((SUM(fs.bytes) - df.bytes\_free) \* 100 / fs.bytes)  
  FROM dba\_temp\_files fs, (SELECT tablespace\_name,bytes\_free,bytes\_used  
          FROM v$temp\_space\_header  
         GROUP BY tablespace\_name,bytes\_free,bytes\_used) df  
 WHERE fs.tablespace\_name (+)  = df.tablespace\_name  
 GROUP BY df.tablespace\_name,fs.bytes,df.bytes\_free,df.bytes\_used  
 ORDER BY 4 DESC;  
**Track all Tablespaces with free space < 10%**  
Select a.tablespace\_name,sum(a.tots/1048576) Tot\_Size, sum(a.sumb/1024) Tot\_Free, sum(a.sumb)\*100/sum(a.tots) Pct\_Free, ceil((((sum(a.tots) \* 15) - (sum(a.sumb)\*100))/85 )/1048576) Min\_Add  
from (select tablespace\_name,0 tots,sum(bytes) sumb  
from dba\_free\_space a  
group by tablespace\_name  
union  
Select tablespace\_name,sum(bytes) tots,0 from dba\_data\_files  
group by tablespace\_name) a group by a.tablespace\_name  
having sum(a.sumb)\*100/sum(a.tots) < 10  
order by pct\_free;  
**Track Tablespace Fragmentation Details:**  
Select a.tablespace\_name,sum(a.tots/1048576) Tot\_Size,  
     sum(a.sumb/1048576) Tot\_Free, sum(a.sumb)\*100/sum(a.tots) Pct\_Free,  
     sum(a.largest/1024) Max\_Free,sum(a.chunks) Chunks\_Free  
     from  ( select tablespace\_name,0 tots,sum(bytes) sumb,  
     max(bytes) largest,count(\*) chunks  
     from dba\_free\_space a  
     group by tablespace\_name  
     union  
     select tablespace\_name,sum(bytes) tots,0,0,0 from dba\_data\_files  
     group by tablespace\_name) a  group by a.tablespace\_name  
order by pct\_free;  
**Track Non-Sys owned tables in SYSTEM Tablespace:**  
SELECT owner, table\_name, tablespace\_name FROM dba\_tables WHERE tablespace\_name = 'SYSTEM' AND owner NOT IN ('SYSTEM', 'SYS', 'OUTLN');  
**Track Default and Temporary Tablespace:**  
SELECT \* FROM DATABASE\_PROPERTIES where PROPERTY\_NAME like '%DEFAULT%';  
select username,temporary\_tablespace,default\_tablespace from dba\_users where username='HRMS';  --for Particular User  
Select default\_tablespace,temporary\_tablespace,username from dba\_users;   --for All Users  
**Track DB datafile used and free space:**  
SELECT SUBSTR (df.NAME, 1, 40) file\_name,dfs.tablespace\_name, df.bytes / 1024 / 1024 allocated\_mb, ((df.bytes / 1024 / 1024) -  NVL (SUM (dfs.bytes) / 1024 / 1024, 0)) used\_mb,  
NVL (SUM (dfs.bytes) / 1024 / 1024, 0) free\_space\_mb  
FROM v$datafile df, dba\_free\_space dfs  
WHERE df.file# = dfs.file\_id(+)  
GROUP BY dfs.file\_id, df.NAME, df.file#, df.bytes,dfs.tablespace\_name  
ORDER BY file\_name;  
**Track Datafile with Archive Details:**  
SELECT NAME, a.status, DECODE (b.status, 'Active', 'Backup', 'Normal') arc, enabled, bytes, change#, TIME ARCHIVE FROM sys.v\_$datafile a, sys.v\_$backup b WHERE a.file# = b.file#;  
**Track Datafiles with highest I/O activity:**  
Select \* from (select name,phyrds, phywrts,readtim,writetim  
from v$filestat a, v$datafile b  
where a.file#=b.file#  
order by readtim desc) where rownum <6;  
**Track Datafile as per the Physical Read/Write Percentage:**  
WITH totreadwrite AS (SELECT SUM (phyrds) phys\_reads, SUM (phywrts) phys\_wrts FROM v$filestat)  
SELECT   NAME, phyrds, phyrds \* 100 / trw.phys\_reads read\_pct, phywrts, phywrts \* 100 / trw.phys\_wrts write\_pct FROM totreadwrite trw, v$datafile df, v$filestat fs WHERE df.file# = fs.file# ORDER BY phyrds DESC;  
**Checking  Autoextend ON/OFF for Datafile:**  
select substr(file\_name,1,50), AUTOEXTENSIBLE from dba\_data\_files  
‎select tablespace\_name,AUTOEXTENSIBLE from dba\_data\_files;  
**More on Tablespace/Datafile size click on the link:**[DB Tablespace/Datafile Details](http://shahiddba.blogspot.com/2012/05/script-to-monitor-tablespacesdatafiles.html)  
**Temp Segment:**  
**Track Temp Segment Free space:**  
SELECT tablespace\_name, SUM(bytes\_used/1024/1024) USED, SUM(bytes\_free/1024/1024) FREE  
FROM   V$temp\_space\_header  
GROUP  BY tablespace\_name;  
SELECT   A.tablespace\_name tablespace, D.mb\_total,  
         SUM (A.used\_blocks \* D.block\_size) / 1024 / 1024 mb\_used,  
         D.mb\_total - SUM (A.used\_blocks \* D.block\_size) / 1024 / 1024 mb\_free  
FROM  v$sort\_segment A, (SELECT   B.name, C.block\_size, SUM (C.bytes) / 1024 / 1024 mb\_total  
         FROM     v$tablespace B, v$tempfile C  
         WHERE    B.ts#= C.ts#  
         GROUP BY B.name, C.block\_size ) D  
WHERE    A.tablespace\_name = D.name  
GROUP by A.tablespace\_name, D.mb\_total;  
**Track Who is Currently using the Temp:**  
SELECT b.tablespace, ROUND(((b.blocks\*p.value)/1024/1024),2)||'M' "SIZE",  
a.sid||','||a.serial# SID\_SERIAL, a.username, a.program  
FROM sys.v\_$session a, sys.v\_$sort\_usage b, sys.v\_$parameter p  
WHERE p.name  = 'db\_block\_size' AND a.saddr = b.session\_addr  
ORDER BY b.tablespace, b.blocks;  
**Undo & Rollback Segment:**  
**Monitor UNDO information:**  
select to\_char(begin\_time,'hh24:mi:ss'),to\_char(end\_time,'hh24:mi:ss'), maxquerylen,ssolderrcnt,nospaceerrcnt,undoblks,txncount from v$undostat  
order by undoblks;  
**Track Active Rollback Segment:**  
SELECT   r.NAME, l.sid, p.spid, NVL (p.username, 'no transaction') "Transaction",  
p.terminal "Terminal" FROM v$lock l, v$process p, v$rollname r  
WHERE l.sid = p.pid(+) AND TRUNC (l.id1(+) / 65536) = r.usn AND l.TYPE(+) = 'TX' AND l.lmode(+) = 6 ORDER BY R.NAME;  
**Track Currently Who is using UNDO and TEMP:**  
SELECT TO\_CHAR(s.sid)||','||TO\_CHAR(s.serial#) sid\_serial,  
 NVL(s.username, 'None') orauser, s.program, r.name undoseg,  
t.used\_ublk \* TO\_NUMBER(x.value)/1024||'K' "Undo"  
FROM sys.v\_$rollname    r, sys.v\_$session s, sys.v\_$transaction t, sys.v\_$parameter   x  
 WHERE s.taddr = t.addr AND r.usn   = t.xidusn(+) AND x.name  = 'db\_block\_size';  
**Redolog Information:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Track Redo Generation by Calender Year:**  
select to\_char(first\_time,'mm.DD.rrrr') day,  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'00',1,0)),'99') "00",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'01',1,0)),'99') "01",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'02',1,0)),'99') "02",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'03',1,0)),'99') "03",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'04',1,0)),'99') "04",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'05',1,0)),'99') "05",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'06',1,0)),'99') "06",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'07',1,0)),'99') "07",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'08',1,0)),'99') "08",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'09',1,0)),'99') "09",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'10',1,0)),'99') "10",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'11',1,0)),'99') "11",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'12',1,0)),'99') "12",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'13',1,0)),'99') "13",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'14',1,0)),'99') "14",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'15',1,0)),'99') "15",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'16',1,0)),'99') "16",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'17',1,0)),'99') "17",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'18',1,0)),'99') "18",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'19',1,0)),'99') "19",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'20',1,0)),'99') "20",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'21',1,0)),'99') "21",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'22',1,0)),'99') "22",  
to\_char(sum(decode(to\_char(first\_time,'HH24'),'23',1,0)),'99') "23"  
from v$log\_history group by to\_char(first\_time,'mm.DD.rrrr')  
order by day;  
**Track Redo generation by day:**  
select trunc(completion\_time) logdate, count(\*) logswitch, round((sum(blocks\*block\_size)/1024/1024)) "REDO PER DAY (MB)" from v$archived\_log  
group by trunc(completion\_time) order by 1;  
**Track How much full is the current redo log file:**  
SELECT le.leseq   "Current log sequence No", 100\*cp.cpodr\_bno/le.lesiz "Percent Full",  
 cp.cpodr\_bno   "Current Block No", le.lesiz   "Size of Log in Blocks"  
FROM x$kcccp cp, x$kccle le  
WHERE le.leseq =CP.cpodr\_seq  
AND bitand(le.leflg,24) = 8;  
**Monitor Running Jobs:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Long Jobs:**  
Select username,to\_char(start\_time, 'hh24:mi:ss dd/mm/yy') started, time\_remaining remaining, message  
from v$session\_longops  
where time\_remaining = 0 order by time\_remaining desc;  
**Monitor Long running Job:**  
SELECT SID, SERIAL#, opname, SOFAR, TOTALWORK,  
ROUND(SOFAR/TOTALWORK\*100,2) COMPLETE  
FROM   V$SESSION\_LONGOPS  
WHERE TOTALWORK != 0 AND SOFAR != TOTALWORK order by 1;  
**Track Long Query Progress in database:**  
SELECT a.sid, a.serial#, b.username , opname OPERATION, target OBJECT,  
TRUNC(elapsed\_seconds, 5) "ET (s)", TO\_CHAR(start\_time, 'HH24:MI:SS') start\_time,  
ROUND((sofar/totalwork)\*100, 2) "COMPLETE (%)"  
FROM v$session\_longops a, v$session b  
WHERE a.sid = b.sid AND b.username not IN ('SYS', 'SYSTEM') AND totalwork > 0  
ORDER BY elapsed\_seconds;  
**Track Running RMAN backup status:**  
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,  
ROUND(SOFAR/TOTALWORK\*100,2) "%\_COMPLETE"  
FROM V$SESSION\_LONGOPS  
WHERE OPNAME LIKE 'RMAN%'  AND OPNAME NOT LIKE '%aggregate%'  
  AND TOTALWORK != 0 AND SOFAR  != TOTALWORK;  
**Monitor Import Rate:**  
Oracle Import Utility usually takes hours for very large tables and we need to track the execution of Oracle Import Process. Below option can help you monitor the rate at which rows are being imported from a running import job.  
select   substr(sql\_text,instr(sql\_text,'into "'),30) table\_name,  
   rows\_processed, round((sysdate-to\_date(first\_load\_time,'yyyy-mm-dd hh24:mi:ss'))\*24\*60,1) minutes,  
   trunc(rows\_processed/((sysdate-to\_date(first\_load\_time,'yyyy-mm-dd hh24:mi:ss'))\*24\*60)) rows\_per\_minute  
from   sys.v\_$sqlarea  
where   sql\_text like 'insert %into "%' and command\_type = 2 and open\_versions > 0;  
**Displays SQL statements for the current database sessions.**  
SELECT s.sid,  s.status,  s.process,   s.schemaname, s.osuser, a.sql\_text,  p.program  
FROM v$session s, v$sqlarea a, v$process p  
WHERE s.SQL\_HASH\_VALUE = a.HASH\_VALUE  
  
AND s.SQL\_ADDRESS = a.ADDRESS AND s.PADDR = p.ADDR;  
**Displays SQL statements that are using the most resources.**  
SELECT \* FROM   (SELECT Substr(a.sql\_text,1,50) sql\_text,  
Trunc(a.disk\_reads/Decode(a.executions,0,1,a.executions)) reads\_per\_execution,  
a.buffer\_gets, a.disk\_reads, a.executions, a.sorts, a.address  
FROM   v$sqlarea a  
ORDER BY 2 DESC)  
  
WHERE  rownum <= &&1;  
**Database SGA Report:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Monitor SGA Information:**  
SELECT SUM(VALUE)/1024/1024 "Size in MB" from SYS.v\_$sga;  
select     NAME,   BYTES from     v$sgastat  order by NAME;  
**Monitor Shared Pool Information:**  
select to\_number(value) shared\_pool\_size, sum\_obj\_size, sum\_sql\_size, sum\_user\_size,  
(sum\_obj\_size + sum\_sql\_size+sum\_user\_size)\* 1.3 min\_shared\_pool  
  from (select sum(sharable\_mem) sum\_obj\_size  
  from v$db\_object\_cache where type <> 'CURSOR'),  
 (select sum(sharable\_mem) sum\_sql\_size from v$sqlarea),  
 (select sum(250 \* users\_opening) sum\_user\_size from v$sqlarea), v$parameter  
 where name = 'shared\_pool\_size';  
**Monitor PGA Information:**  
Select st.sid "SID", sn.name "TYPE", ceil(st.value / 1024 / 1024/1024) "GB"  
from v$sesstat st, v$statname sn where st.statistic# = sn.statistic#  
and sid in (select sid from v$session where username like UPPER('hrms'))  
and upper(sn.name) like '%PGA%' order by st.sid, st.value desc;  
**Monitor CPU Usage Information:**  
select  ss.username, se.SID, VALUE/100 cpu\_usage\_seconds  
from v$session ss,  v$sesstat se,  v$statname sn where se.STATISTIC# = sn.STATISTIC#  
and NAME like '%CPU used by this session%' and se.SID = ss.SID  
and  ss.status='ACTIVE' and  ss.username is not null order by VALUE desc;  
**Disk I/O Report:**  
WITH totreadwrite AS (SELECT SUM (phyrds) phys\_reads, SUM (phywrts) phys\_wrts FROM v$filestat)  
SELECT   NAME, phyrds, phyrds \* 100 / trw.phys\_reads read\_pct,  
    phywrts, phywrts \* 100 / trw.phys\_wrts write\_pct  
 FROM totreadwrite trw, v$datafile df, v$filestat fs  
   WHERE df.file# = fs.file# ORDER BY phyrds DESC;  
**IO Usage for a Query:**  
select b.sql\_text "Statement ", a.Disk\_reads "Disk Reads", a.executions "Executions",  
a.disk\_reads/decode(a.executions,0,1,a.executions) "Ratio",c.username  
from  v$sqlarea a, v$sqltext\_with\_newlines b,dba\_users c  
where  a.parsing\_user\_id = c.user\_id and a.address=b.address and a.disk\_reads>100000  
order by a.disk\_reads desc,b.piece;  
**Display the System write batch size:**  
SELECT kviival write\_batch\_size  
  FROM x$kvii  
 WHERE kviidsc = 'DB writer IO clump' OR kviitag = 'kcbswc'  
**Monitor Disk I/O Contention:**  
select   NAME,  PHYRDS "Physical Reads",  
    round((PHYRDS / PD.PHYS\_READS)\*100,2) "Read %",   PHYWRTS "Physical Writes",  
    round(PHYWRTS \* 100 / PD.PHYS\_WRTS,2) "Write %",   fs.PHYBLKRD+FS.PHYBLKWRT "Total Block I/O's" from (    select     sum(PHYRDS) PHYS\_READS, sum(PHYWRTS) PHYS\_WRTS  
    from    v$filestat    ) pd,  v$datafile df,  v$filestat fs  
where     df.FILE# = fs.FILE#  
order     by fs.PHYBLKRD+fs.PHYBLKWRT desc;  
For information about database latch statistics and wait information. Click on the below link: [**Latch Statistics & Wait information**](http://shahiddba.blogspot.com/2013/08/script-to-monitor-top-latches.html)  
**DB Locks/Blocks/Blocker Details:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Track Block session in oracle 9i/10g**  
‎select s1.username || '@' || s1.machine || ' ( SID=' || s1.sid ||  ' )  is blocking ' || s2.username || '@' || s2.machine || ' ( SID=' ||  s2.sid || ' ) ' AS blocking\_status from gv$lock l1, gv$session s1, gv$lock l2, gv$session s2 where s1.sid = l1.sid and s2.sid = l2.sid  and l1.BLOCK = 1  and l2.request > 0  and l1.id1 = l2.id1  and l2.id2 = l2.id2;  
select do.object\_name, row\_wait\_obj#, row\_wait\_file#, row\_wait\_block#, row\_wait\_row#,  
dbms\_rowid.rowid\_create(1, ROW\_WAIT\_OBJ#, ROW\_WAIT\_FILE#, ROW\_WAIT\_BLOCK#, ROW\_WAIT\_ROW#)  
from gv$session s, dba\_objects do  
where sid = 543 and s.ROW\_WAIT\_OBJ# = do.OBJECT\_ID;  
*For detail description of blocking you can run this on your Oracle-Home*  
**oracle-home\rdbms\admin\utllockt.sql**  
Select process,sid, blocking\_session from v$session where blocking\_session is not null;  --in 10g  
**Track Locked Session & Blocked:**  
PROMPT Blocked and Blocker Sessions  
select /\*+ ORDERED \*/ blocker.sid blocker\_sid, blocked.sid blocked\_sid ,  
TRUNC(blocked.ctime/60) min\_blocked, blocked.request  
from (select \*from v$lock  
where block != 0 and type = 'TX') blocker, v$lock blocked  
where blocked.type='TX' and blocked.block = 0 and blocked.id1 = blocker.id1;  
**Track Database Lock:**  
Select /\*+ ORDERED \*/ l.sid, l.lmode,  
TRUNC(l.ctime/60) min\_blocked, u.name||'.'||o.NAME blocked\_obj  
from (select \* from v$lock  
where type='TM' and sid in (select sid  
from v$lock where block!=0)) l, sys.obj$ o, sys.user$ u  
where o.obj# = l.ID1 and o.OWNER# = u.user#;  
**Track the Session Waiting for Lock:**  
SELECT holding\_session bsession\_id, waiting\_session wsession\_id, b.username busername, a.username wusername, c.lock\_type TYPE, mode\_held, mode\_requested, lock\_id1, lock\_id2  
FROM sys.v\_$session b, sys.dba\_waiters c, sys.v\_$session a  
WHERE c.holding\_session = b.sid AND c.waiting\_session = a.sid;  
**Track Blocker Details:**  
SELECT sid, serial#, username, osuser, machine  
FROM v$session  
WHERE sid IN (select sid from v$lock  
where block != 0 and type = 'TX');  
**Users/Sessions/Processes Details:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Average Wait Time for Particular Event:**  
SELECT EVENT,  TOTAL\_WAITS,  TOTAL\_TIMEOUTS,  TIME\_WAITED, round(AVERAGE\_WAIT,2) "Average Wait"  
 from v$system\_event order    by TOTAL\_WAITS;  
**Sessions Waiting On A Particular Wait Event:**  
SELECT count(\*), event  
FROM v$session\_wait  
WHERE wait\_time = 0 AND event NOT IN ('smon timer','pipe get','wakeup time manager', 'pmon timer','rdbms ipc message', 'SQL\*Net message from client')  
GROUP BY event ORDER BY 1 DESC;  
**Track Logon time of DB user and OS user:**  
Select to\_char(logon\_time,'dd/mm/yyyy hh24:mi:ss'),osuser,status,schemaname,machine from v$session where type !='BACKGROUND'; ‎  
**Track all Session User Details:**  
select sid, serial#,machine, status, osuser,username from v$session where username!='NULL';  
**Track Active Session User Details:**  
SELECT SID, Serial#, UserName, Status, SchemaName, Logon\_Time FROM V$Session WHERE Status= 'ACTIVE' AND UserName IS NOT NULL;  
**Track Active User Details:**  
SELECT s.inst\_id,  s.sid,  s.serial#,  p.spid,  s.username,  s.program FROM gv$session s  JOIN gv$process p ON p.addr = s.paddr AND p.inst\_id = s.inst\_id WHERE s.type != 'BACKGROUND';  
**Report OS Process ID for each session:**  
SELECT    ses.username  || '('  || ses.sid  || ')' users, acc.owner owner, acc.OBJECT OBJECT, ses.lockwait, prc.spid os\_process  
  FROM v$process prc, v$access acc, v$session ses  
 WHERE prc.addr = ses.paddr AND ses.sid = acc.sid;  
**Show Username and SID/SPID with Program Name:**  
select sid,name,value from v$spparameter where isspecified='TRUE';‎  
SELECT SID, Serial#, UserName, Status, SchemaName, Logon\_Time FROM V$Session  
WHERE Status= 'ACTIVE' AND UserName IS NOT NULL;  --to find active session  
SELECT s.inst\_id,  s.sid,  s.serial#,  p.spid,  s.username,  s.program    --active users details  
FROM gv$session s  JOIN gv$process p ON p.addr = s.paddr AND p.inst\_id = s.inst\_id  
WHERE s.type != 'BACKGROUND';  
**Track Current Transaction in Database:**  
‎‎select a.sid, a.username, b.xidusn, b.used\_urec, b.used\_ublk  from v$session a, v$transaction b  
where a.saddr = b.ses\_addr;‎  
**Important Object Information:**  
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  
**Database Object Information:**  
Select owner,object\_type,count(\*) from dba\_objects Where owner not IN ('SYS','MDSYS','CTXSYS','HR','ORDSYS','OE','ODM\_MTR','WMSYS','XDB','QS\_WS', 'RMAN','SCOTT','QS\_ADM','QS\_CBADM', 'ORDSYS','OUTLN','PM','QS\_OS','QS\_ES','ODM','OLAPSYS','WKSYS','SH','SYSTEM','ORDPLUGINS','QS','QS\_CS')  
Group by owner,object\_type order by owner;  
**Query to Find 5 largest object in Database:**  
SELECT \* FROM (select SEGMENT\_NAME, SEGMENT\_TYPE, BYTES/1024/1024/1024 GB, TABLESPACE\_NAME from dba\_segments order by 3 desc ) WHERE ROWNUM <= 5;  
**Track Last DDL Performed in database:**  
Select CREATED, TIMESTAMP, last\_ddl\_time from all\_objects WHERE OWNER='HRMS' AND OBJECT\_TYPE='TABLE' order by timestamp desc;  
**Count Invalid Object:**  
Select owner, object\_type, count(\*) from dba\_objects where status='INVALID' group by  owner, object\_type;  
**Report all Invalid Object in Database:**  
SELECT owner, object\_name, object\_type,‎ TO\_CHAR (last\_ddl\_time, 'DD-MON-YY hh:mi:ss') last\_time FROM dba\_objects‎ WHERE status = 'INVALID';  
**Report Invalid Object with Next Action:**  
select 'Alter ' || decode(object\_type,'PACKAGE BODY','PACKAGE',object\_type) || ' ' || object\_name || ' compile ' || decode(object\_type,'PACKAGE BODY',' body;',';') from user\_objects where object\_type in ('FUNCTION','PACKAGE','PACKAGE BODY','PROCEDURE','TRIGGER','VIEW') and status = 'INVALID' order by object\_type , object\_name;  
Click on the link to Report Invalid object and How to Compile them**:**[**Report All Invalid Objects**](http://shahiddba.blogspot.com/2012/07/script-to-report-all-invalid-objects-in.html)

**Track Total Number of Table/Index/Mviews:**

Select count(1) from user\_tables where table\_name not like '%$%'  
Select count(1) from user\_mviews;  
Select count(1) from user\_indexes where index\_type in ('FUNCTION-BASED NORMAL','NORMAL');  
**Number of Objects Created in last week:**  
Select count(1) from user\_objects where CREATED >= sysdate - 7  
**Track Mviews Not Refreshed since last Week:**  
Select mview\_name from user\_mviews where LAST\_REFRESH\_DATE < sysdate - 7;