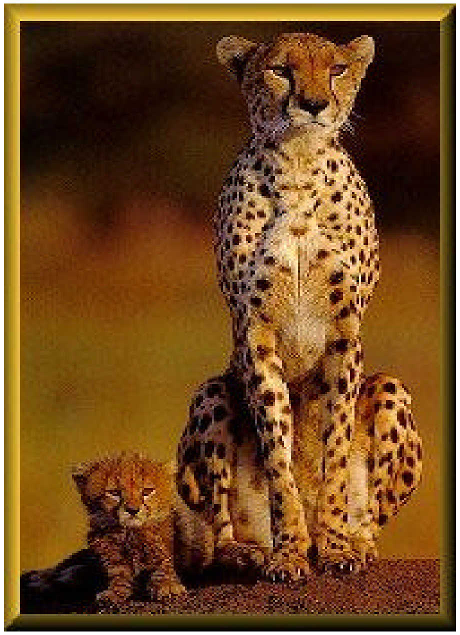
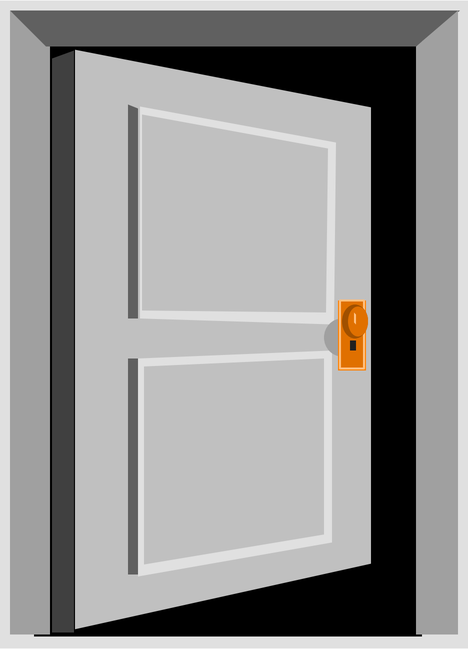
V$ Views – Don’t Leave $HOME Without Them



Mike Bell

BMC Software



Dos and Don'ts

- **Will Do**
  - **Look at some V$ View Basics**
  - **Talk about the underlying X$ Tables**
  - **Categorize The V$ Views**
  - **Uncover the undocumented init.ora parameters**
  - **Present some useful scripts that are based on the V$ Views**
  - **Learn about a wild animal**
- **Won't Do**
  - **Provide a complete reference of the V$ Views**
  - **Show an explain tree (won't even mention it)**
  - **Turn this into a tuning class**
  - **Bore the experienced DBA (hopefully not)**
  - **Traumatize the new DBA (definitely not)**

Presentation Objectives

- Review V$ View Fundamentals
- V$ View Creation and Access Methods
- Categorize the V$ Views
- Provide Useful V$ View Scripts
  - Memory Allocation
  - Problem Queries
  - Media Recovery
- Show the Value of Each Script

V$View Fundamentals

What Are The V$ Views?

- Unfiltered, unbiased looks into the heart of the Oracle database
- Supplement to the data dictionary
- Basis for database performance monitoring and tuning
- Key to moving from the average to the expert DBA

► V$ View Fundamentals

How Are V$ Views Created?

- V$ Views are created from the X$ tables when the database is created
- The CATALOG.SQL and CATLDR.SQL scripts are executed
- A V_$ view is created from the V$ view
- Old synonyms for the V$ views are dropped
- New V$ synonyms are created on the V_$ view

► # V$ View Fundamentals

How's That Again?

The V$ Views that are accessed by SYSTEM are actually synonyms that point to the V_$ Views that are views of the original V$ Views based on the X$ Tables!

V$View Fundamentals

Here’s an excerpt from the CATALOG.SQL Script

```
create or replace view v_$database as select * from v$database;

drop public synonym v$database;

create public synonym v$database for v_$database;

grant select on v_$database to select_catalog_role;
```

► V$ View Fundamentals

How Are They Accessed?

- SELECT access only
- Can’t grant access to V$ views even as SYS
- You can grant access to the underlying V_$ views

***Tip – It’s better to grant access to the V_$ Views to a specific user than to give users the SYS or SYSTEM passwords.**

V$View Fundamentals

What Are X$ Tables?

- X$ tables are fixed tables created in memory at database startup
- Store up-to-date information on database activity
- Cannot be dropped
- Cannot be updated (except by Oracle)
- Only accessible by SYS

► V$View Fundamentals

The X$ V$ Connection

- X$ tables are very cryptic
- V$ views created on X$ tables for readability
- Several X$ tables and columns are not referenced by V$ views
- V$ views are created on one or more X$ tables

V$View Fundamentals

The X$ V$ Connection (cont’d)

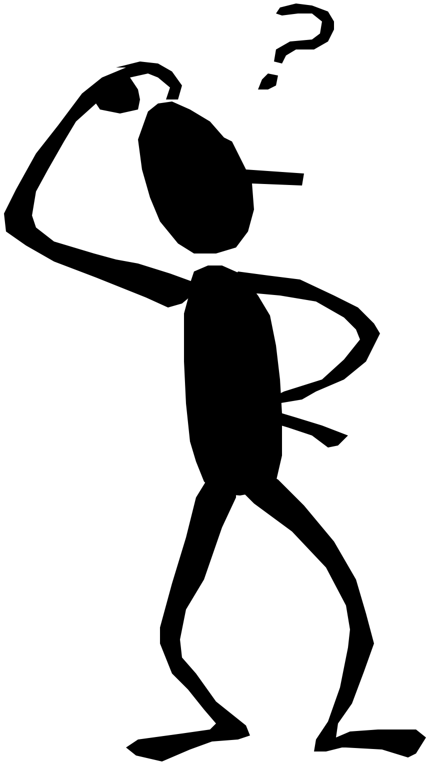
- X$ tables and V$ views are only accessible by SYS
- 183 V$ views and 265 X$ tables in Oracle 8.1.6.0.0
- 227 V$ views and 352 X$ tables in Oracle 9.0.1.1.1

V$View Fundamentals

Listing the X$ Tables and V$ Views

```
select type,name from
v$fixed_table order by type,name;
```

```
TYPE  NAME
----- ------------------------------
TABLE X$ACTIVECKPT
TABLE X$BH
VIEW  GV$ACCESS
VIEW  GV$ACTIVE_INSTANCES
VIEW  V$ACCESS
VIEW  V$ACTIVE_INSTANCES
```



V$View Fundamentals

X$ Tables That Make up The V$ Views

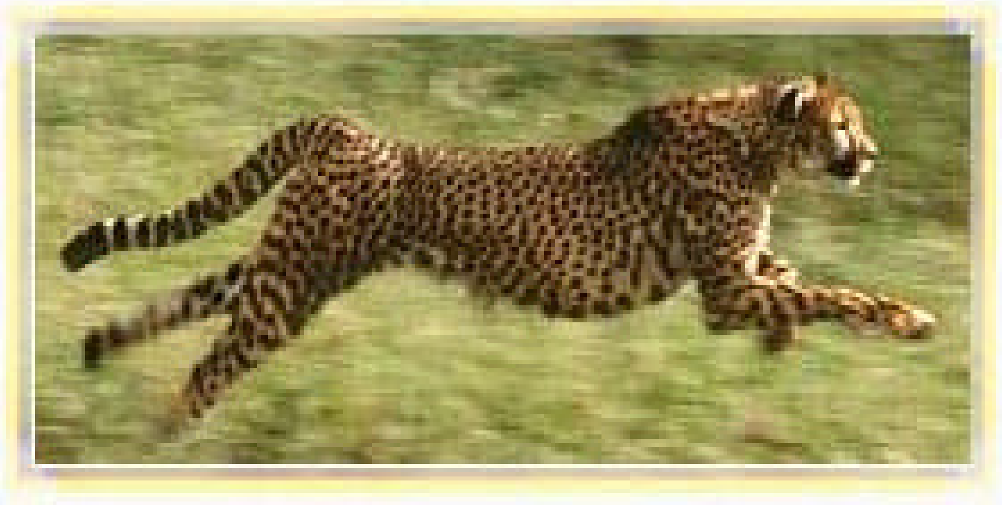
```
select * from v$fixed_view_definition where
view_name ='GV$FIXED_TABLE';
```

```
VIEW_NAME               VIEW_DEFINITION
----------------------  --------------------------------------------------------
GV$FIXED_TABLE          select inst_id,kqftanam, kqftaobj, 'TABLE', indx
                        from x$kqfta
                        union all
                        select inst_id,kqfvinam, kqfviobj, 'VIEW', 65537
                        from x$kqfvi
                        union all
                        select inst_id,kqfdtnam, kqfdtobj, 'TABLE',
                        65537 from x$kqfdt
```

► Cheetah Factoid

Cheetahs Can Run a Sustained Speed of?

- 25 miles-an-hour
- 45 miles-an-hour
- 65 miles-an-hour
- 100 miles-an-hour



65 miles-an-hour

V$View Categories

- **Backups, Archiving, And Recovery**
  - **V$BACKUP, V$ARCHIVE, V$RECOVER_FILE, etc.**
- **Caches**
  - **V$ROWCACHE, V$LIBRARYCACHE, etc.**
- **Control Files**
  - **V$CONTROLFILE, V$CONTROLFILE_RECORD_SECTION**
- **SQL Statements and Cursors**
  - **VSQL, VSQLAREA, V$SQLTEXT, V$OPEN_CURSOR, etc.**
- **Database/Instance**
  - **V$DATABASE, V$INSTANCE, V$VERSION, etc.**
- **SQL*Loader Direct Path Load Option**
  - **V$LOADCSTAT, V$LOADPSTAT, V$LOADSTAT**

V$View Categories

- Fixed Views
  - V$FIXED_TABLE, V$FIXED_VIEW_DEFINITION
- General
  - V$TIMER, V$TYPE_SIZE, V$_SEQUENCES
- I/O
  - V$FILESTAT, V$WAITSTAT
- Latches/Locks
  - V$BUFFER_POOL, V$LATCH_MISSES, V$LOCK, etc.
- MTS/OPS
  - V$CIRCUIT, V$DISPATCHER, V$QUEUE, etc.
- Overall System Performance
  - V$SYSTAT, V$SORT_USAGE, etc.

V$View Categories

- Parallel Query
  - V$EXECUTION, V$PQ_SESSTAT, etc.
- Oracle Parameters
  - V$PARAMETER, V$NLS_PARAMETERS, etc.
- Redo Logs
  - VLOG, VLOGFILE, V$LOGHIST
- Rollback Segments
  - V$ROLLSTAT, V$TRANSACTION
- Security/Privileges
  - V$ENABLED_PRIVS, V$PWFILE_USERS
- Sessions
  - V$SESSION, V$PROCESS, etc.

V$View Fundamentals

282 Undocumented Parameters in 8.1.6.0.0

432 Undocumented Parameters in 9.0.1.1.1

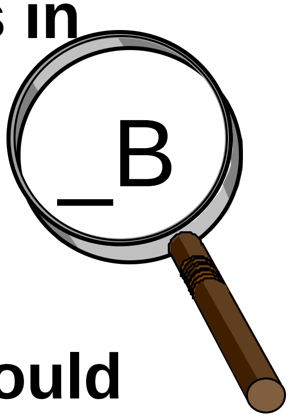
```
select indx,ksppinm from x$ksppi where
substr(ksppinm,1,1) = '_';
```

```
 INDX     KSPPINM
--------- ----------------------------
      0   _trace_files_public
      2   _latch_miss_stat_sid
      3   _max_sleep_holding_latch
      4   _max_exponential_sleep
    190   _corrupted_rollback_segments
```

Warning Warning Warning

Oracle's Warning

Warning: Information about the dynamic performance views is presented for completeness only; this information does not imply a commitment to support these views in the future.



Murphy's Warning

Warning: All undocumented parameters should be extensively tested before and after use.

► Goal #1: Memory Allocation

The Goal is to Ensure That The Correct Amount of Memory is Allocated to Oracle.

Oracle SGA

The RDBMS buffer and working storage area

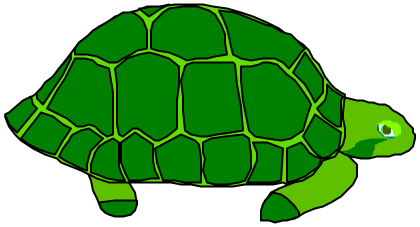
Contains buffers for:

- Data Buffers
- Redo Buffers
- Data Dictionary / SQL / Procedures / Packages information

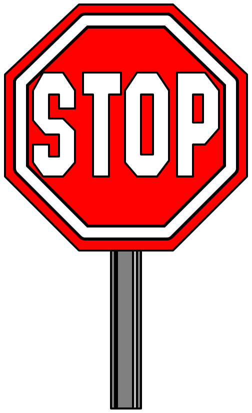
Created at instance startup

► DB_BLOCK_BUFFERS:

If DB_BLOCK_BUFFERS is low, users will not have enough memory to operate efficiently.



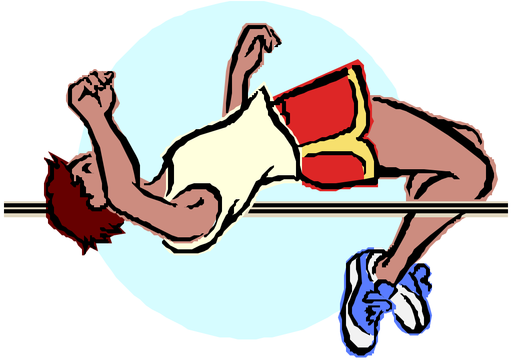
If DB_BLOCK_BUFFERS is high, your system may begin to swap and may come to a halt.



► Determine if DB_BLOCK_BUFFERS Is Set Too Low:

```
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 "Read Hit Ratio"
from   v$sysstat;
```

```
Read Hit Ratio
98.415926
```



Use X$bh to Get State of SGA

```
select      state, count(*)
from        x$bh
group by    state;
```

STATE	COUNT(*)	Status (fyi)
0	920	free
1	1553	available
3	27	being used

DB_BLOCK_BUFFERS available = 1553
DB_BLOCK_BUFFERS being used = 27
DB_BLOCK_BUFFERS never used = 920

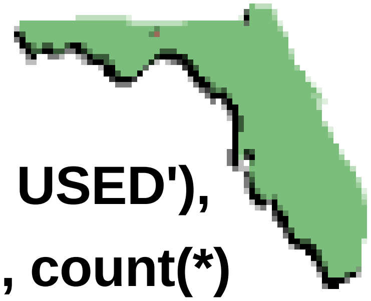
Hold On A Minute!
There's Some Misleading Information Here!

```
select      state, count(*)
from        x$bh
where state = 1
and         lrba_seq <> 0
Group by state;
```

```
COUNT
   38
```

Some blocks listed at state = 1 can still be in use by an existing query!

X$bh - More accurate query!



```
Select
  decode(state,0, 'FREE',
                 1, decode(lrba_seq,0,'AVAILABLE','BEING USED'),
                 3, 'BEING USED', state) "BLOCK STATUS", count(*)
From x$bh
group by decode(state,0,'FREE',
                     1,decode(lrba_seq,0, 'AVAILABLE','BEING USED'),
                     3, 'BEING USED', state);
```

```
BLOCK STATUS      COUNT(*)
AVAILABLE         1515
BEING USED        65
FREE              920
```

► Shared SQL Area/Shared Pool

Contains:

- **Pre-parsed database procedures**
- **Pre-parsed database triggers**
- **Recently parsed SQL**
- **Recently parsed PL/SQL requests**

The SHARED_POOL_SIZE:

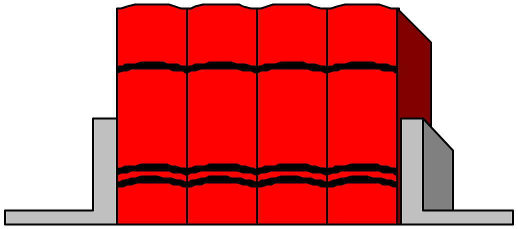
This is the memory allocated for the library and data dictionary cache

If the SHARED_POOL_SIZE is set too low then you will not get the full advantage of your DB_BLOCK_BUFFERS.

Determine library cache hit ratio.

```
select   sum(pins) Executions, sum(pinhits) “Execution Hits”,
         ((sum(pinhits) / sum(pins)) * 100) phitrat,
         sum(reloads) Misses,
         ((sum(pins) / (sum(pins) + sum(reloads))) * 100)  hitrat
from     v$librarycache;
```

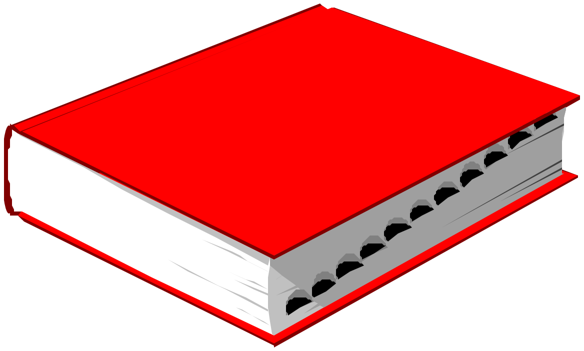
```
Executions Execution Hits   PHITRAT      Misses  HITRAT
   12810        11651       90.952381      48     99.6266
```



► Dictionary Buffers

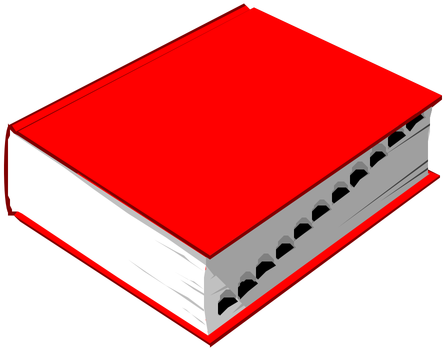
Contain:

- ◆Table definitions
- ◆Column definitions
- ◆View definitions
- ◆Sequence definitions
- ◆Role definitions
- ◆Authorizations & Permissions



Data dictionary cache miss ratio:

```
select sum(gets) “Gets”,
       sum(getmisses) “Misses”,
       (1 - (sum(getmisses) / (sum(gets) +
       sum(getmisses))))*100 “HitRate”
from   v$rowcache;
```



```
Gets          Misses        HitRate
10233            508        95.270459
```

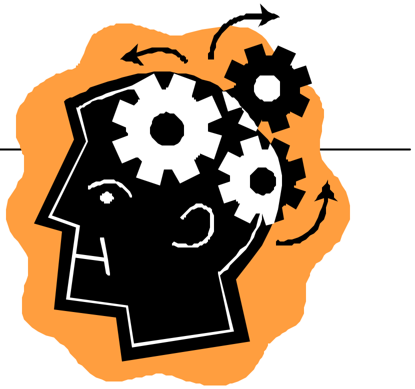
This would be a good Ratio and would probably not require action in this area.

Determine the Memory Left in the SHARED_POOL_SIZE:

```
col value for 999,999,999,999 heading “Shared Pool Size”
col bytes for 999,999,999,999 heading “Free Bytes”
select   to_number(v$parameter.value) value, v$sgastat.bytes,
         (v$sgastat.bytes/v$parameter.value)*100 “Percent Free”
from     v$sgastat, v$parameter
where    v$sgastat.name = 'free memory'
and      v$parameter.name = 'shared_pool_size'
and     v$sgastat.pool = 'shared pool';
```

```
Shared Pool Size   Free Bytes         Percent Free
20,480,000         12,778,732          62.3961523
```

Determine the Memory Left in the SHARED_POOL_SIZE:



```
select *
from   v$sgastat
where name = 'free memory'
 and pool = 'shared pool';
```

```
POOL          NAME                          Free Bytes
shared pool free memory                     12,759,956
```

If there is free memory on a consistent basis, then there is no need to increase this parameter.

Memory Left - SHARED POOL:

```
select sum(ksmchsiz) Bytes, ksmchcls Status
from   x$ksmsp
group by ksmchcls;

   Free Bytes        STATUS
----------------     --------
   1,024,000         R-free
          40         R-freea
   3,176,368         free
   6,130,576         freeabl
  11,631,584         perm
   1,297,996         recr
```

What Do These STATUS Values Mean?

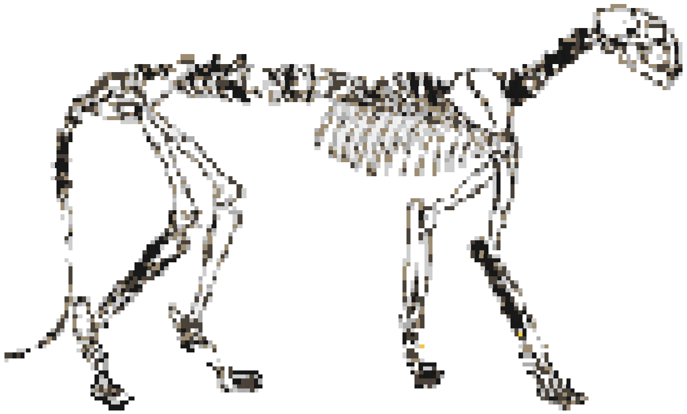
Oracle does not state what the STATUS values mean. Here is a table of possible descriptions courtesy of Rich Niemiec.

r-free	This is SHARED_POOL_RESERVER_SIZE
r-freea	This is probably reserved memory that has been used but freeable
free	This is the amount of contiguous free memory available
freeabl	This is probably memory that has been used but is freeable
perm	This is free memory not yet moved to the free area for use
recr	Not quite sure – possibly reserved memory for Oracle

Cheetah Factoid

The Oldest Fossils of The Cheetah (from about 10,000 years ago) were found where?

- North America
- Asia
- Africa
- Europe



MIRACINONYX INEXPECTATUS
prehistoric cheetah-like cat
of North America

► Goal #2: Identify Problem Queries

The Goal is to Identify SQL Statements That Are Causing Database Performance Degradation.

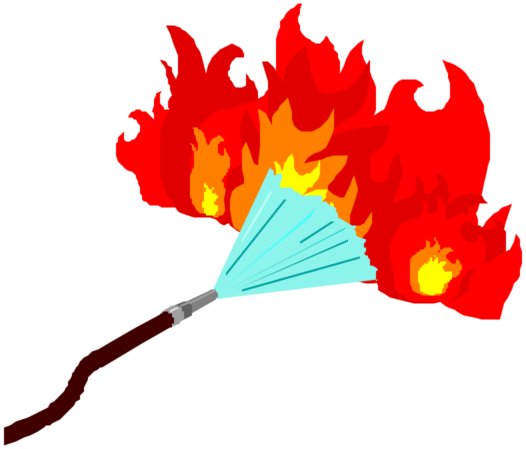
Identifying Which Statements Are The Top Disk Readers (Physical)

```
Select disk_reads, sql_text
from v$sqlarea
Where disk_reads > 10000
order  by disk_reads desc;
```

DISK READS	SQL TEXT
12,987	select order#,columns,types from orders where substr(orderid,1,2)=:1
11,131	select custid, city from customer where city = ‘DETROIT’

Identifying Which Statements Are The Top Buffer Readers (Logical)

```
select buffer_gets, sql_text
from v$sqlarea
Where buffer_gets > 200000
order by buffer_gets desc;
```



```
Buffer_gets    SQL_TEXT
300,219        select order#,cust_no, from
               orders where division = '1'
```

Finding The Hot Procedures

1. Set Up The Report Headings

```
col PTYP format a15 heading 'PL/SQL Type'
col OBJ  format a40 wrap heading 'Name'
col EXES format 999,990  heading 'Execs'
```

Finding The Hot Procedures (cont’d)

2. Select From The V$ Views

```
select PTYP, OBJ , 0 - EXEM  EXES
  from (select distinct EXEM, PTYP, OBJ
          from ( select O.TYPE PTYP, O.OWNER || '.'
                   || O.NAME OBJ,
                   0 - O.EXECUTIONS EXEM
                   from  V$DB_OBJECT_CACHE  O
```

Finding The Hot Procedures (cont’d)

3. Filter The Results

```
where O.TYPE in ( 'FUNCTION', 'PACKAGE',
          'PACKAGE BODY', 'PROCEDURE' ,
          'TRIGGER'
                       ) ) )
 where ROWNUM <= &1
```

Finding the Hot Procedures (cont'd)

4. Execute The Query

```
@hotprocs.sql   5
```

```
PL/SQL  Type    Name                                Execs

PACKAGE              SYS.DBMS_APPLICATION_INFO      267
PACKAGE    BODY      SYS.DBMS_APPLICATION_INFO      266
PACKAGE              SYS.DBMS_OUTPUT                 39
PACKAGE    BODY      SYS.DBMS_OUTPUT                 38
PACKAGE              SYS.DBMS_SPACE                   1
```

Finding the Hot Tables

1. Set Up The Report Headings

```
col CTYP heading 'Command Type'
col OBJ  format a32 wrap   heading 'Table'
col EXES format 999,990    heading 'Execs'
col GETS format 99,999,990 heading 'Buff Gets'
col ROWP format 99,999,990 heading 'Rows Proc'
```

Finding the Hot Tables (cont’d)

2. Select From The V$ Views

```
select CTYP, OBJ, 0 - EXEM  EXES, GETS, ROWP
  from (select distinct EXEM, CTYP, OBJ, GETS, ROWP
          from ( select decode (S.COMMAND_TYPE
                                   ,  2, 'Insert into ',  3, 'Select from '
                                   ,  6, 'Update  of  ',  7, 'Delete from '
                                   , 26, 'Lock    of  ') CTYP
                            , O.OWNER || '.' || O.NAME    OBJ
                            , sum(0 - S.EXECUTIONS)       EXEM
                            , sum(S.BUFFER_GETS)          GETS
                            , sum(S.ROWS_PROCESSED)       ROWP
 from V$SQL S, V$OBJECT_DEPENDENCY D
                          , V$DB_OBJECT_CACHE   O
```

Finding the Hot Tables (cont’d)

3. Filter The Results

```
where S.COMMAND_TYPE in (2,3,6,7,26)
                  and D.FROM_ADDRESS = S.ADDRESS
                  and D.TO_OWNER     = O.OWNER
                  and D.TO_NAME      = O.NAME
                  and O.TYPE         = 'TABLE'
              group by S.COMMAND_TYPE
                    , O.OWNER
                    , O.NAME )  )
 where ROWNUM <= &1
/
```

Finding the Hot Tables (cont’d)

```
@hottbls.sql   10
```

Command Type	Table	Execs	Buff Gets	Rows Proc
Select from	SYS.USER$	2,122	895,172	25,485
Select from	SYS.OBJ$	2,005	730,363	21,538
Insert into	BMC.MIG_ROWS	1,000	1,140	1,000
Update of	BMC.MIG_ROWS	1,000	53,260	1,000
Insert into	BMC.ITEM	880	1,878	880
Insert into	BMC.EMPLOYEE	608	1,312	608
Select from	SYS.CDEF$	584	1,771	232
Select from	SYS.IND$	558	180,488	4,756
Select from	SYS.JOB$	516	572	0
Select from	SYS.TAB$	510	275,021	8,010

Finding Miscellaneous Stats

Hot Stats STATS.SQL

```
col NAME  format a55 heading 'Statistic'
col VALUE format 999,999,990 heading 'Value'
select NAME, VALUE  from V$SYSSTAT
  where NAME like '%&1%‘ order by NAME
```

Finding Miscellaneous Stats (cont’d)

STATS.SQL

```
@stats.sql redo
```

Statistic	Value
redo blocks written	82
redo buffer allocation retries	0
redo entries	147
redo entries linearized	0
redo log space requests	0
redo log space wait time	0

► Finding Miscellaneous Stats (cont’d)

STATS.SQL

```
@stats.sql sort
```

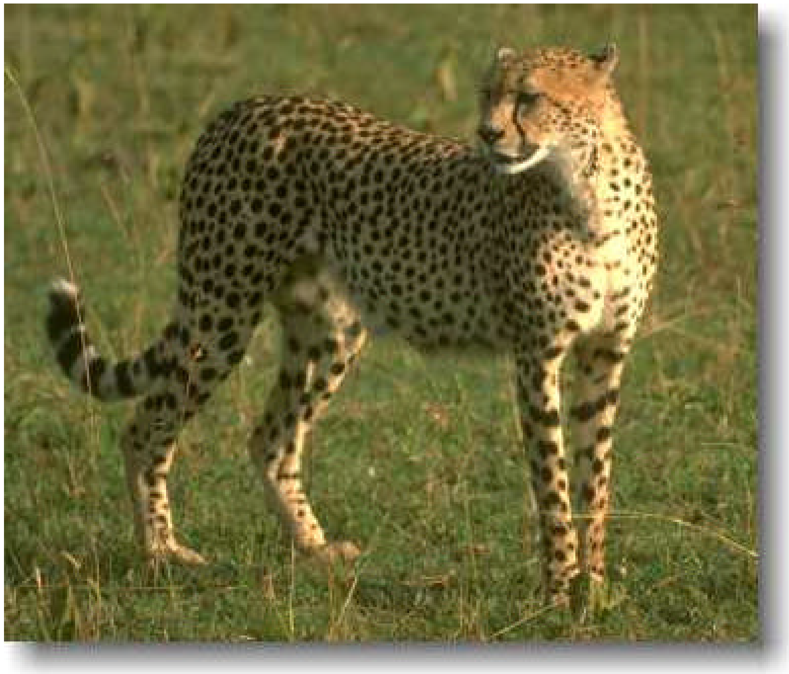
```
Statistic                          Value

sorts  (disk)                          0
sorts  (memory)                      120
sorts  (rows)                        739
```

Cheetah Factoid

Of The 12,400 Cheetahs Remaining in The World, How Many Remain In The Wild?

- **1,000**
- **2,400**
- **5,000**
- **12,400**



2,400

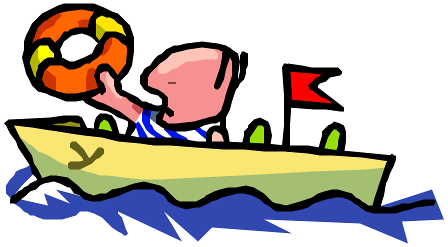
► Goal #3 - Media Recovery

Have You Ever Gotten This Error?

```
SQL> startup
ORACLE instance started.

Total System Global Area   65004812 bytes
Fixed Size                    70924 bytes
Variable Size              44376064 bytes
Database Buffers           20480000 bytes
Redo Buffers                  77824 bytes
Database mounted.
ORA-01113: file 11 needs media recovery
ORA-01110: data file 11: D:\ORACLE\ORADATA\B816\SFI_DEBUG_1.DBF'
```

► Media Recovery (cont’d)

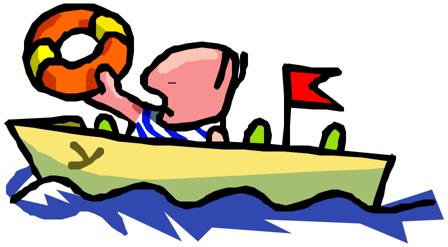


Some people restore the database multiple times after receiving this error and still can not resolve the problem.

What has happened is the one or more of the tablespaces were in backup mode when the database was shutdown.

Since the tablespace was in BACKUP mode, the datafile SCN number is not consistent with the last SCN of the database.

Media Recovery (cont’d)

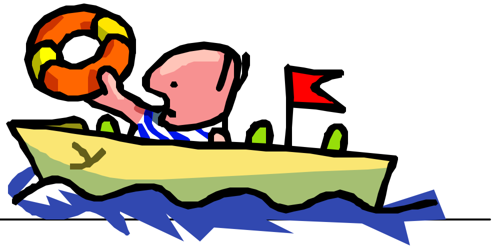


1. **Query The V$BACKUP & V$DATAFILE Views to Identify The Datafiles.**

```
SQL> SELECT NAME, V$BACKUP.STATUS
  2  FROM V$BACKUP, V$DATAFILE
  3  WHERE V$BACKUP.FILE# =  V$DATAFILE.FILE#
  4  and V$BACKUP.STATUS = 'ACTIVE';

NAME                                              STATUS
D:\ORACLE\ORADATA\B816\SFI_DEBUG_1.DBF             ACTIVE
```

Media Recovery (cont’d)

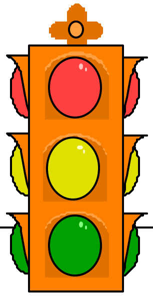


2. Generate The Alter Database Statement

For Multiple Files:

```
select 'ALTER DATABASE DATAFILE ''' || name
   ||
''' END BACKUP;'
from v$backup, v$datafile
where v$backup.file# = v$datafile.file#
and v$backup.status = 'ACTIVE’;
```

Media Recovery (cont’d)



3. Execute The Alter Database Statement

```
SQL> alter database
        datafile 'D:\ORACLE\ORADATA\B816\SFI_DEBUG_1.DBF'
        end backup;
Database altered.
SQL> alter database open;
Database altered.
```

Here's the Plug…

DBXray

- Real-time Performance Monitoring

SQL Explorer

- Collect/Tune/Test SQL Statements

Space Expert

- Identify Storage Problems/Recommend and Implement Solutions

PATROL for Oracle

- 24x7 Database Monitoring

SQL Programmer

- SQL Development

SQL Backtrack

- Efficient Backups and Guided Recoveries

Here’s the Credits

Oracle Performance Tuning Tips & Techniques

Expert Technical Knowledge at the Turn of a Page. Tune and tailor you Oracle7, or Oracle8 system to perform at its peak using this one-of-a-kind reference, designed for every Oracle database administrator, programmer, and user.

► Thanks for Coming!

About the Presenter:

Mike Bell is a Technical Product Manager for Oracle for BMC Software, a major provider of Enterprise Monitoring and Management software. Mike can be reached at michael_bell@bmc.com.

References:

Oracle Design; Dave Ensor & Ian Stevenson; O'REILLY
Oracle Performance Tuning Tips & Techniques; Richard J. Niemiec; Oracle Press
Oracle PL/SQL Tips and Techniques, Joseph C. Trezzo; Oracle Press
"Oracle Scripts Every DBA Should Know" by Joseph C. Trezzo; TUSC
"The V$ Views; A DBA's Best Friend" by Joseph C. Trezzo; TUSC
"Performance Tuning for the Expert; Others Will Require Oxygen!" by Richard J. Niemiec; TUSC
Cheetah Conservation Fund: The WILD Foundation PO Box 1380 Ojai, CA, USA e-mail - info@cheetah.org

Neither BMC Software nor the presenter guarantee this document to be error-free. The opinions expressed in this presentation are the presenter's and may not reflect the opinions of BMC Software or of any living person. Please provide comments and/or questions to michael_bell@bmc.com.