SELECT® Online
The Journal of the International Oracle Users Group

Select Magazine - July 1998
Volume 5, No. 4

The V$ Views - A DBA's Best Friend

By Joseph C. Trezzo

This paper was presented at ***IOUG-A Live! '98***, May 10-15 in Orlando. This presentation received an overall score of 3.90 on a scale of 5.0, earning it Best Paper Honors for this year's conference. To view all of the papers from this year's conference, visit the IOUG-A web site at www.ioug.org.

There are a set of views that supplements the Oracle data dictionary that contain real-time information about the current state of the database and various facets of the database and the database environment. This information can be very helpful in performance tuning, recovery, resource contention, and identification.

The information is stored in the SGA and updated similar to the method in which the data dictionary is updated, when an action takes place, it typically updates one of these views underlying table(s). These views are known as the fixed tables, virtual tables, v$ tables, dynamic performance tables, and a half a dozen other names. They are commonly referred to as tables, but remember they are views. The x$ tables, underlying tables of the v$ views, are maintained in memory real-time and store up to date information on the current activity of the database at the current point in time or since the last database startup.

This paper will focus specifically on the v$ views and on the education of these tables and the powerfulness of their applicability. Once an understanding of these tables is reached, the valuable information obtainable is limitless . This paper will contain several segments of source code and will be supplemented by several additional examples in the presentation.

V$ View Creation and Access

These views are created by the *catalog.sql* script, and as of Oracle8 there are approximately 130 views created (actually two of the views are created by the *catldr.sql* script that is used for SQL*Loader direct load statistical information), all with the prefix of "v_$". The underlying view creations for each v$ view can be seen in the v$ view name v$fixed_view_definition. The views are created by selecting from one or more x$ tables. In the same script, there is a view created for each "v_$" view to allow users to access the view. Users cannot access the "v_$" views and therefore, this method provides access to these views via a view on a view. The view name changes the prefix of each view to "v$". And lastly, there is a public synonym created on each view since the tables are owned by the ***SYS*** user. An example of a v$ view creation in the *catalog.sql* script is shown below:

```
create or replace view v_$datafile as
       select * from v$datafile;
drop   public synonym v$datafile;
create public synonym v$datafile for v_$datafile;
```

The x$ tables, underlying tables of the v$ views, are virtual tables, which are created in memory at database startup and maintained in memory. They are in every sense treated as tables with some limitations. These tables cannot be modified in any manner and no indexes can be created on these tables. The only operation that can be performed on these tables and views is a SELECT. In order to provide access to the v$ views. Since the v$ views are accessible to a user that belongs to the DBA role, I have created a script called *gntmontr.sql* to grant privilege to a specified user on all the v$ views, thus if you want someone to have access to the v$ views and not the DBA type privileges, this is possible. I also created a *revmontr.sql* script to revoke all the v$ view access from a specified user as well. Keep in mind, three roles are created when the database in created for compatibility with version 6, namely, connect, resource and dba. These roles are created by the *sql.bsq* file and this section of code is displayed below.

```
create role connect;
grant create session,alter session,create synonym, create view,
   create database link,create table,create cluster,
   create sequence to connect;
create role resource;
grant create table,create cluster,create sequence,
     create trigger, create procedure to resource;
create role dba;
grant all privileges to dba with admin option;
```

Please note that Oracle explicitly states in their manuals that these views may change in the future and as evidenced from the modifications from the version 6 to version 7 and version 7 to version 8, this is definitely true, not only have there been an increase in the number of views, but some of the views have changed columns as well.

V$ View Categories

The views are categorized according to their primary function and are often times needed to join to another category to retrieve certain information. Below is a description of each category followed by the actual v$ views that belong to each group.

Category Descriptions

Backup/Archive/Recovery: Information related to database backups, archive files, and recovery, including last backup, archive logs, state of files for backup, and recovery.

Caches: Information related to the various caches, including objects, library, cursors, and the dictionary.

Control Files: Information related to control files.

Cursors/SQL Statements: Information related to cursors and SQL statements, including the open cursors, statistics, and actual SQL text.

Databases/Instances: Information related to database and instance configuration and setup.

Direct Loader: Information related to the SQL*Loader direct load option.

Fixed Views: Information related to the v$ views.

I/O: Information related to I/O, including files, and statistics.

Latches/Locks: Information related to latches and locks.

Miscellaneous: Information related to views that do not fall into a category.

Multi-Threaded/Parallel Server: Information related to multi-threaded and parallel servers, including connections, queues, dispatchers, and shared servers.

Overall System: Information related to the overall system performance.

Parallel Query: Information related to the parrallel query option.

Parameters: Information related to various Oracle parameters, including initialization and nls per session.

Redo Logs: Information related to redo logs, including statistics and history.

Rollback Segments: Information on rollback segments, including statistics and transactions.

Security/Privileges: Information related to security.

Sessions: Information related to a session, including object access, cursors, processes, and statistics.

Views per Category

BACKUPS/ARCHIVES/RECOVERY

V$ARCHIVE V$ARCHIVED_LOG (V8)
V$ARCHIVE_DEST (V8) V$BACKUP
V$BACKUP_CORRUPTION (V8) V$BACKUP_DATAFILE (V8)
V$BACKUP_DEVICE (V8) V$BACKUP_PIECE (V8)
V$BACKUP_REDOLOG (V8) V$BACKUP_SET (V8)
V$DELETED_OBJECT (V8) V$RECOVERY_FILE_STATUS
V$RECOVERY_LOG V$RECOVERY_STATUS
V$RECOVER_FILE

CACHES

V$CACHE (V8) V$DB_OBJECT_CACHE
V$LIBRARYCACHE V$ROWCACHE
V$SUBCACHE (V8)

CONTROL FILES

V$CONTROLFILE V$CONTROLFILE_RECORD_SECTION (V8)

CURSORS/SQL STATEMENTS

V$OPEN_CURSOR V$SQL
V$SQLAREA V$SQLTEXT
V$SQLTEXT_WITH_NEWLINES V$SQL_BIND_DATA (V8)
V$SQL_BIND_METADATA (V8) V$SQL_CURSOR
V$SQL_SHARED_MEMORY

DATABASES/INSTANCES

V$ACTIVE_INSTANCES V$BGPROCESS
V$BH (V8) V$COMPATIBILITY
V$COMPATSEG V$COPY_CORRUPTION (V8)
V$DATABASE V$DATAFILE
V$DATAFILE_COPY (V8) V$DATAFILE_HEADER (V8)
V$DBFILE V$DBLINK
V$DB_PIPES V$INSTANCE
V$LICENSE V$OFFLINE_RANGE (V8)
V$OPTION V$SGA
V$SGASTAT V$TABLESPACE (V8)
V$VERSION

DIRECT LOADER

V$LOADCSTAT V$LOADPSTAT (V8)
V$LOADTSTAT

FIXED VIEWS

V$FIXED_TABLE V$FIXED_VIEW_DEFINITION
V$INDEXED_FIXED_COLUMN

I/O

V$FILESTAT V$WAITSTAT

LATCHES/LOCKS

V$BUFFER_POOL (V8) V$CACHE_LOCK
V$CLASS_PING (V8) V$DLM_CONVERT_LOCAL (V8)
V$DLM_CONVERT_REMOTE (V8) V$DLM_LATCH (V8)
V$DLM_MISC (V8) V$ENQUEUE_LOCK (V8)
V$EVENT_NAME V$FALSE_PING
V$FILE_PING (V8) V$LATCH
V$LATCHHOLDER V$LATCHNAME
V$LATCH_CHILDREN V$LATCH_MISSES
V$LATCH_PARENT V$LOCK
V$LOCK_ACTIVITY V$LOCK_ELEMENT
V$LOCKED_OBJECT V$LOCKS_WITH_COLLISIONS
V$PING V$RESOURCE
V$RESOURCE_LIMIT (V8) V$TRANSACTION_ENQUEUE (V8)
V$_LOCK V$_LOCK1

MISCELLANEOUS

V$TIMER V$TYPE_SIZE
V$_SEQUENCES (V8)

MULTI-THREADED/PARALLEL SERVER

V$CIRCUIT V$DISPATCHER
V$DISPATCHER_RATE (V8) V$MTS
V$QUEUE V$REQDIST
V$SHARED_SERVER V$THREAD

OVERALL SYSTEM

V$GLOBAL_TRANSACTION (V8) V$OBJECT_DEPENDENCY
V$SHARED_POOL_RESERVED V$SORT_SEGMENT
V$SORT_USAGE (V8) V$STATNAME
V$SYSSTAT V$SYSTEM_CURSOR_CACHE
V$SYSTEM_EVENT V$TRANSACTION

PARALLEL QUERY

V$EXECUTION V$EXECUTION_LOCATION (V8)

V$PQ_SESSTAT V$PQ_SLAVE

V$PQ_SYSSTAT V$PQ_TQSTAT

PARAMETERS

V$NLS_PARAMETER V$NLS_VALID_VALUES
V$PARAMETER V$SYSTEM_PARAMETER

REDO LOGS

V$LOG V$LOGFILE
V$LOGHIST V$LOG_HISTORY

ROLLBACK SEGMENTS

V$ROLLNAME V$ROLLSTAT

SECURITY/PRIVILEGES

V$ENABLEDPRIVS V$PWFILE_USERS

SESSIONS

V$ACCESS V$MYSTAT
V$PROCESS V$SESSION
V$SESSION_CONNECT_INFO V$SESSION_CURSOR_CACHE
V$SESSION_EVENT V$SESSION_LONGOPS (V8)
V$SESSION_OBJECT_CACHE (V8) V$SESSION_WAIT
V$SESSTAT V$SESS_IO

The new v$ views in Oracle8 are designated with a "(V8)" after each view name. Also, one other important note, some of the v$ timing columns are dependent on the timed_statistics parameter in the *init.ora* file being set to true, otherwise there will be no timing in these columns.

Sample Queries

This section is centered around the "educate by example" method of education and therefore, I will display several queries with a brief explanation of the information that is retrieved from a query of this type. This should provide a very good knowledge base on the major v$ views and the information that is readily available in these views.

Determine dispatcher activity, if very busy, add more dispatchers.

```
select name, network, status, accept, idle, busy
from   v$dispatcher;
```

Determine dispatcher queue average wait time.

```
select paddr, type, queued, wait,
totalq,
          wait/totalq avg_wait_time
from   v$queue;
```

Determine shared server activity, if idle often, remove a shared server(time converted to minutes).

```
select name, status, idle/6000, busy/6000, requests
from   v$shared_server;
```

Determine library cache hit ratio. If the hit ratio or reloads is high, increase the shared_pool_size parameter in the init.ora file.

```
select namespace, gethitratio, pinhitratio,
reloads
from   v$librarycache;
```

Determine dictionary cache hit ratio, if > 15%, increase the shared_pool_size init.ora parameter. The dc SGA storage is now a part of the shared pool along with the library cache, there are no more "dc_" parameters in the init.ora. This is a key area since the dictionary is accessed so frequently especially by the internals of Oracle. I would never recommend lowering the shared_pool_size since the library cache is also a part of this shared pool.

```
select sum(gets), sum(getmisses),
          sum(getmisses)/sum(gets) "Ratio of Misses"
from   v$rowcache;
```

Determine if the data block buffers are being utilized optimally. If the read hit ratio is less than 95%, increasing the db_block_buffers may help performance. However, beware that if the ratio is near 100% and the number of gets is in the millions, there is a very good chance that the statement being executed is not optimized. Keep in mind, the hit ratio could be near 100% and a query could take very long to complete execution. Likewise, the hit ratio could be much less than 100% and a query could take a very short time to complete execution. The key is to also consider the gets or logical reads, if the number is extremely high, then examine the application code.

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

Determine the I/O that is taking place on each database file. If the reads and writes are not distributed evenly between files, the tablespaces may need to be restructured for better performance.

```
select a.file#, a.name, a.status, a.bytes, b.
phyrds,
          b.phywrts
from   v$datafile a, v$filestat b
where  a.file# = b.file#;
```

Display rollback information and determine if more segments are needed. If the waits to gets goes over one, then add more rollback segments.

```
select a.name, b.extents, b.rssize, b.xacts, b.
waits,
          b.gets, optsize,status
from   v$rollname a, v$rollstat b
where  a.usn = b.usn;
```

Determine the number of sessions for each user.

```
select username, count
(*)
from    v$session
group by username;
```

Display detail information about a session, including the operating system username and process, and terminal.

```
select sid, username, program, osuser,
process,
          machine, terminal, type
from   v$session;
```

Display detailed statistics per session to determine a session's resource usage.

```
select a.sid, a.username, b.name, c.value
from   v$session a, v$statname b, v$sesstat c
where  a.sid = c.sid
and      b.statistic# = c.statistic#
and      a.username = upper('&username')
order by a.sid, a.username, b.name;
```

Display the current cursor being executed by a session.

```
select a.sid, a.username, b.sql_text
from   v$session a, v$open_cursor b
where  a.saddr = b.saddr;
```

Display the entire sql statement being executed by a session.

```
select a.sid, a.username, b.sql_text
from   v$session a, v$sqltext b
where a.sql_address = b.address
and     a.sql_hash_value = b.hash_value
order by a.sid, a.username, b.piece;
```

Display each cursor of the shared cursor cache, including the times executed and loaded into the cache.

```
select sql_text, version_count, executions, loads
from   v$sqlarea;
```

Determine the objects that are being locked by each session.

```
select a.sid, a.username, b.owner, b.object, b.ob_typ
from   v$session a, v$access b
where  a.sid = b.sid;
```

Display the processes for each session. If the process is a background process, then the operating system and identifier are retrieved from the process table.

```
select a.sid,
        decode(b.background,1,b.program,a.username)
        "user", b.pid,
        decode(b.background,1,b.spid,a.audsid) "os id"
from   v$session a, v$process b
where  a.paddr = b.addr;
```

Display each rollback segment that has current transactions, along with the session id, serial#, username, and actual sql statement being executed in the rollback segment. A session can be killed by using the system kill command (alter system kill session 'sid,serial#'; where sid and serial# are obtained from the v$session view).

This allows a DBA to determine which rollback segment is processing each transaction.

```
select a.name, b.xacts, c.sid, c.serial#, c.
username,
          d.sql_text
from   v$rollname a, v$rollstat b, v$session c,
          v$sqltext d,v$transaction e
where  a.usn = b.usn
and      b.usn = e.xidusn
and      c.taddr = e.addr
and      c.sql_address = d.address
and      c.sql_hash_value = d.hash_value
order by a.name, c.sid, d.piece;
```

Display database information, including the name, creation date, and archive mode (on or off).

```
select *
from   v$database;
```

Display the actual memory in bytes that each init.ora parameter is using. The v$sga table summarizes the bytes by the type of parameter.

```
select *
from   v$sgastat;
```

Display details on the redo log files, including the filename, size, and archive status.

```
select a.member, b.*
from   v$logfile a, v$log b
where  a.group# = b.group#;
```

Display a history of the archive logs created along with the file names.

```
select *
from   v$log_history;
```

Display the natural language parameters in effect for a session. A user can alter the different parameters for a session. The default date format is "dd-mon-yy". If a user wants to change this for the current session, they

could enter the command "alter session set nls_date_format='mm/dd/yy'" and all dates returned in that session would display in that format.

```
select *
from   v$nls_parameters;
```

Display all the init.ora parameters and the current values, along with signifying if the value is the default value.

```
select name, value, isdefault
from   v$parameter
order by name;
```

Display licensing information, including the maximum number of licensed concurrent or named users, the current number of active concurrent sessions, and the highest number of users that were concurrently logged on at anyone time since the last database startup.

```
select sessions_max, sessions_current,
          sessions_highwater, users_max
from   v$license;
```

Determine the memory usage per session.

```
select a.sid, a.username, b.value
from   v$session a, v$sesstat b, v$statname c
where  a.sid = b.sid
and      b.statistic# = c.statistic#
and      c.name = 'session memory';
```

This by no means is all encompassing, there is much more that can be obtained from these tables, but the intent is to provide a solid starting point.

Summary

This papers intent was to provide an education on the v$ views, how they are created, what they are used for, and how they can be utilized by DBAs to take advantage of this valuable information that Oracle provides. They are truly an extension to the standard data dictionary and should be treated as such. If you have any questions about this paper, feel free to call me at 630-960-2909. In addition, this paper will be downloadable from the TUSC home web page as www.tusc.com. There is a poster that contains the v$ view definitions grouped by category that can be obtained from TUSC by calling TUSC and requesting a V$ poster. This poster is current as of Oracle8.

References

Oracle7 Server: SQL Language Reference Manual, Oracle Corporation
Oracle7 Server: Application Developer's Guide, Oracle Corporation

A special thanks to All TUSC staff for their help in putting this paper together.
