

PRACTICAS TEMA 1.

ARQUITECTURA DE LA BASE DE DATOS.

- 1.1. Comprobar las variables de entorno necesarias para conectarnos a la BD.
- 1.2. Identificar los procesos que componen instancia.
- 1.3. Ver el tamaño de la SGA de la BD (v\$sgainfo).
- 1.4. Comprobar valores de parámetros del init relacionados con el tamaño de la SGA.
- 1.5. Comprobar ficheros que componen la BD y ubicarlos en la estructura OFA.
- 1.6. Identificar la estructura lógica de la BD: tablespaces, segmentos, extensiones.
- 1.7. Consultar información sobre la base de datos (v\$database) y la instancia (v\$instance).
- 1.8. Localizar el proceso “servidor” asociado a mi sesión (v\$process y v\$session). ¿Es un servidor dedicado o compartido?
- 1.9. ¿Cuanto ocupa la Dictionary cache y la Library cache en tu BD? (v\$sgastat)
- 1.10. Ver la actividad de la Library Cache (v\$librarycache).
- 1.11. Ver las sentencias SQL que guarda la Shared-Pool (v\$sqlarea).
- 1.12. Crear un fichero de autenticación y activar su uso.
- 1.13. Asignar la variable NLS_LANG para nuestro país y comprobar el cambio en las respuestas de Oracle desde sqlplus.
- 1.14. Subir el tamaño de la shared-pool un gránulo más y comprobar cómo aumenta el espacio libre.
- 1.15. Comprobar el funcionamiento de la caché de redolog, como protectora del contenido de la caché de datos. Para ello iniciaremos una transacción y provocaremos una caída de la BD, comprobando que al arrancarla de nuevo, se mantendrá la integridad de la misma.
- 1.16. Comprobar el funcionamiento de la caché de datos, en lo que se refiere a la mejora del rendimiento cuando se repite una consulta. ¿Por qué la segunda vez que se lanza la misma consulta tarda menos?

En primer lugar, si ya no lo está, hay que arrancar la base de datos. En el ejemplo vemos el arranque de la base de datos CURSOxy (siempre q se haga referencia a la BD CURSOxy, sustituye xy por los números q incluya el nombre de tu usuario linux):

```
/home/CURSO/cursoXY (CURSOxy)> sqlplus
SQL*Plus: Release 10.2.0.1 - Production on Wed Oct 20 12:39:05 2006
Copyright (c) 1982, 2005, Oracle Corporation. All rights reserved.
Enter user-name: / as sysdba
Connected to an idle instance.
```

Si ya está arrancada nos dará un error:

```
SQL> startup
ORA-01081: no se puede iniciar ORACLE cuando ya se esta ejecutando - cierrelo primero
```

Si no lo está:

```
SQL> startup
Instancia ORACLE iniciada.
```

```
Total System Global Area      83886080 bytes
Fixed Size                      1259288 bytes
Variable Size                   62916840 bytes
Database Buffers                16777216 bytes
Redo Buffers                     2932736 bytes
Base de datos montada.
Base de datos abierta.
```

1.1. Comprobar la asignación de variables de entorno necesarias para conectarnos a la BD.

Se trata de las variables \$ORACLE_HOME, \$ORACLE_SID, \$LD_LIBRARY_PATH y \$PATH. ORACLE_HOME define en qué directorio está instalado Oracle. ORACLE_SID determina con qué instancia queremos trabajar. LD_LIBRARY_PATH permite que Oracle localice las librerías compartidas que no forman parte del núcleo. PATH debe incluir el directorio con los ejecutables de Oracle, para mayor comodidad del administrador.

Nota: si no asignamos LD_LIBRARY_PATH en Linux, es posible que no podamos ejecutar sqlplus, obteniendo un error similar a:

```
sqlplus: error while loading shared libraries: libclntsh.so: cannot open shared
object file: No such file or directory
```

Si no sucede el error es posible que no se haya compilado con librerías compartidas, o que el sistema tenga acceso a dichas librerías de otro modo, como /etc/ld.so.conf o un enlace en /usr/lib, etc.

Solución:

```
/home/CURSO/cursoXY (CURSOxy)> echo $ORACLE_HOME
/u01/app/oracle/product/10.2.0.1
/home/CURSO/cursoXY (CURSOxy)> echo $ORACLE_SID
CURSOxy
/home/CURSO/cursoXY (CURSOxy)> echo $PATH
/u01/app/oracle/product/10.2.0.1/bin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/CURSO/cursoXY/bin
/home/CURSO/cursoXY (CURSOxy)> echo $LD_LIBRARY_PATH
/u01/app/oracle/product/10.2.0.1/lib
```

1.2. Identificar los procesos que componen instancia.

Los podemos identificar desde el S.O., por ejemplo, en Unix con el comando "ps -ef" que permite ver todos los procesos en ejecución en el sistema, filtrando por el nombre de la instancia de BD:

```
ps -ef|grep CURSOxy
```

También lo podemos hacer consultando la vista dinámica V\$PROCESS (o haciendo join con V\$BGPROCESS, que contiene todos los posibles procesos background), en la BD en cuestión:

```
select * from v$process where background is not null;
```

o bien

```
select * from v$process a, v$bgprocess b where a.ADDR=b.PADDR;
```

Solución:

```
/home/CURSO/cursosXY (CURSOxy)> ps -ef|grep CURSOxy|grep -v grep
oracle  22515      1  0 19:29 ?          00:00:00 ora_pmon_CURSOxy
oracle  22517      1  0 19:29 ?          00:00:00 ora_psp0_CURSOxy
oracle  22519      1  0 19:29 ?          00:00:00 ora_mman_CURSOxy
oracle  22521      1  0 19:29 ?          00:00:00 ora_dbw0_CURSOxy
oracle  22523      1  0 19:29 ?          00:00:00 ora_lgwr_CURSOxy
oracle  22525      1  0 19:29 ?          00:00:00 ora_ckpt_CURSOxy
oracle  22527      1  0 19:29 ?          00:00:00 ora_smon_CURSOxy
oracle  22529      1  0 19:29 ?          00:00:00 ora_reco_CURSOxy
oracle  22531      1  0 19:29 ?          00:00:00 ora_mmon_CURSOxy
oracle  22533      1  0 19:29 ?          00:00:00 ora_mmnl_CURSOxy
oracle  22538      1  0 19:29 ?          00:00:00 ora_qmnc_CURSOxy
oracle  22544      1  0 19:29 ?          00:00:00 ora_q000_CURSOxy
oracle  22551      1  0 19:30 ?          00:00:00 ora_q001_CURSOxy
```

```
SQL> select username,program from v$process
       where background is not null;
```

USERNAME	PROGRAM
cursoXY	oracle@cursos.atika.um.es (PMON)
cursoXY	oracle@cursos.atika.um.es (PSP0)
cursoXY	oracle@cursos.atika.um.es (MMAN)
cursoXY	oracle@cursos.atika.um.es (DBW0)
cursoXY	oracle@cursos.atika.um.es (LGWR)
cursoXY	oracle@cursos.atika.um.es (CKPT)
cursoXY	oracle@cursos.atika.um.es (SMON)
cursoXY	oracle@cursos.atika.um.es (RECO)
cursoXY	oracle@cursos.atika.um.es (MMON)
cursoXY	oracle@cursos.atika.um.es (MMNL)
cursoXY	oracle@cursos.atika.um.es (QMNC)
cursoXY	oracle@cursos.atika.um.es (q000)
cursoXY	oracle@cursos.atika.um.es (q001)

13 filas seleccionadas.

```
SQL> select name,description from v$process a, v$bgprocess b
       where a.ADDR=b.PADDR;
```

NAME	DESCRIPTION
PMON	process cleanup

Administración Básica de Oracle10g

```
PSP0 process spawner 0
MMAN Memory Manager
DBW0 db writer process 0
LGWR Redo etc.
CKPT checkpoint
SMON System Monitor Process
RECO distributed recovery
MMON Manageability Monitor Process
MMNL Manageability Monitor Process 2
QMNC AQ Coordinator
11 filas seleccionadas.
```

1.3. Ver el tamaño de la SGA de la BD y las cachés que la componen.

Hay varias vistas dinámicas de la BD que nos dan información sobre el tamaño y la estructura de la SGA: V\$SGAINFO, V\$SGA_DYNAMIC_COMPONENTS, V\$SGA_TARGET_ADVICE, V\$SGA y V\$SGASTAT. Las tres primeras son nuevas en 10g.

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

Solución:

(Con Oracle 10g, en V\$SGAINFO podemos ver cómo Oracle reparte el espacio de la SGA (sga_target) entre todas las cachés que la componen, así como el tamaño del gránulo y el tamaño máximo al que podría llegar la SGA (sga_max_size))

```
SQL> select * from v$sgainfo;
NAME                                BYTES RES
-----
Fixed SGA Size                       1259528 No
Redo Buffers                          2932736 No
Buffer Cache Size                     46137344 Yes
Shared Pool Size                      46137344 Yes
Large Pool Size                       4194304 Yes
Java Pool Size                        4194304 Yes
Streams Pool Size                      0 Yes
Granule Size                          4194304 No
Maximum SGA Size                     125829120 No
Startup overhead in Shared Pool      29360128 No
Free SGA Memory Available            20971520
11 filas seleccionadas.
```

(En v\$sgastat también podemos ver información detallada sobre la Shared Pool)

```
SQL> select * from v$sgastat;
POOL      NAME                                BYTES
-----
          fixed_sga                       1259528
          buffer_cache                     46137344
          log_buffer                       2932736
          ...
shared pool sql area                   2206900
          ...
shared pool library cache              2561104
          ...
shared pool free memory                 9928264
          ...
shared pool row cache                   3741868
```

Administración Básica de Oracle10g

...

587 filas seleccionadas.

```
SQL> select * from v$sgastat
       where name in ('library cache','row cache','sql area',
                    'buffer_cache','log_buffer');
```

POOL	NAME	BYTES
	buffer_cache	46137344
	log_buffer	2932736
shared pool	sql_area	2206900
shared pool	library cache	2561104
shared pool	row cache	3741868

1.4. Comprobar valores de parámetros del init relacionados con el tamaño de la SGA.

Los parámetros de inicialización más importantes que afectan al tamaño de la SGA son: `shared_pool_size`, `db_cache_size`, `db_block_size`, `log_buffer`, `large_pool_size` y `java_pool_size`. Con Oracle 10g se introduce `sga_target` para que, automáticamente, se ajuste el tamaño de las cachés que componen la SGA, nunca por encima de `sga_max_size`. Por tanto, con 10g, bastaría asignar `sga_target`, `sga_max_size` y `log_buffer` (y los demás a cero, aunque en la práctica lo mejor es asignarles un valor mínimo).

Solución:

```
SQL> show parameter db_block_size
```

NAME	TYPE	VALUE
db_block_size	integer	2048

```
SQL> show parameter sga_target
```

NAME	TYPE	VALUE
sga_target	big integer	100M

```
SQL> show parameter sga_max_size
```

NAME	TYPE	VALUE
sga_max_size	big integer	120M

```
SQL> show parameter log_buffer
```

NAME	TYPE	VALUE
log_buffer	integer	2886656

```
SQL> show parameter shared_pool_size
```

NAME	TYPE	VALUE
shared_pool_size	big integer	0

```
SQL> show parameter db_cache_size
```

NAME	TYPE	VALUE
db_cache_size	big integer	0

```
SQL> show parameter large_pool_size
```

NAME	TYPE	VALUE
------	------	-------

Administración Básica de Oracle10g

```
-----  
large_pool_size                big integer 0
```

```
SQL> show parameter java_pool_size  
NAME                           TYPE          VALUE  
-----  
java_pool_size                  big integer 0
```

(Si comprobamos los tamaños que Oracle realmente ha asignado a las cachés, veremos algunas diferencias con los parámetros iniciales, relacionadas con los valores mínimos de algunas cachés, así como con el posible ajuste automático que haya hecho el propio Oracle 10g)

```
SQL> select * from v$sgainfo;
```

```
NAME                           BYTES RES  
-----  
Fixed SGA Size                  1259528 No  
Redo Buffers                    2932736 No  
Buffer Cache Size              46137344 Yes  
Shared Pool Size               46137344 Yes  
Large Pool Size                4194304 Yes  
Java Pool Size                 4194304 Yes  
Streams Pool Size              0 Yes  
Granule Size                   4194304 No  
Maximum SGA Size               125829120 No  
Startup overhead in Shared Pool 29360128 No  
Free SGA Memory Available      20971520
```

11 filas seleccionadas.

(La columna RES, de resizeable, vale "Yes" para aquellas partes de la SGA cuyo tamaño es gestionado automáticamente por Oracle)

```
SQL> select rpad(component,30),CURRENT_SIZE,USER_SPECIFIED_SIZE,min_size  
from V$SGA_DYNAMIC_COMPONENTS;
```

```
RPAD(COMPONENT,30)             CURRENT_SIZE USER_SPECIFIED_SIZE MIN_SIZE  
-----  
shared pool                    50331648      0      29360128  
large pool                     4194304        0         0  
java pool                      4194304        0      4194304  
streams pool                   0              0         0  
DEFAULT buffer cache          41943040      0      41943040  
KEEP buffer cache              0              0         0  
RECYCLE buffer cache          0              0         0  
DEFAULT 2K buffer cache        0              0         0  
DEFAULT 4K buffer cache        0              0         0  
DEFAULT 8K buffer cache        0              0         0  
DEFAULT 16K buffer cache       0              0         0  
DEFAULT 32K buffer cache       0              0         0  
ASM Buffer Cache               0              0         0
```

13 filas seleccionadas.

1.5. Comprobar ficheros que componen la BD y ubicarlos en la estructura OFA.

Los ficheros que componen la BD los podemos consultar desde el propio S.O. o desde la BD. Desde el S.O., y teniendo en cuenta la estructura OFA los encontraremos en /u0?/oradata/CURSOxy, dado que nuestros puntos de montaje son /u01, /u02, /u03 y /u04:

```
ls -l /u0?/oradata/CURSOxy
```

Administración Básica de Oracle10g

Podemos localizar cada tipo de fichero (datafile, control y redolog) por la extensión; por ejemplo: dbf, ctl y log.

¡¡¡ Ojo !!!, ésto no es más que una convención, las extensiones pueden ser otras, o incluso no existir.

Otra opción es acceder a la información de los ficheros de control desde la propia BD, consultando las vistas dinámicas V\$DATAFILE, V\$TEMPFILE, V\$CONTROLFILE y V\$LOGFILE:

```
select * from v$datafile;
select * from v$tempfile;
select * from v$logfile;
select * from v$controlfile;
```

Solución:

```
(Podemos ver los ficheros de la BD desde el S.O.)
/home/CURSO/cursoXY (CURSOxy)> ls -l /u0?/oradata/$ORACLE_SID
/u02/oradata/CURSOxy:
total 285632
-rw-rw---- 1 oracle dba 9060352 feb 27 13:28 control1.ctl
-rw-rw---- 1 oracle dba 272631808 feb 27 13:27 system01.dbf
-rw-rw---- 1 oracle dba 10487808 feb 27 13:13 users01.dbf

/u03/oradata/CURSOxy:
total 90676
-rw-rw---- 1 oracle dba 9060352 feb 27 13:28 control2.ctl
-rw-rw---- 1 oracle dba 52430848 feb 27 13:13 sysaux01.dbf
-rw-rw---- 1 oracle dba 10487808 feb 26 15:38 temp01.dbf
-rw-rw---- 1 oracle dba 20973568 feb 27 13:27 undo_rbs01.dbf

/u04/oradata/CURSOxy:
total 30780
-rw-rw---- 1 oracle dba 10486272 feb 27 13:12 redo01.log
-rw-rw---- 1 oracle dba 10486272 feb 27 13:27 redo02.log
-rw-rw---- 1 oracle dba 10486272 feb 27 13:12 redo03.log
```

(También podemos ver qué ficheros componen la BD, desde la propia BD, mediante consultas SQL)

```
SQL> select name from v$datafile;
NAME
```

```
-----
/u02/oradata/CURSOxy/system01.dbf
/u03/oradata/CURSOxy/undo_rbs01.dbf
/u03/oradata/CURSOxy/sysaux01.dbf
/u02/oradata/CURSO22/users01.dbf
```

```
SQL> select name from v$tempfile;
NAME
```

```
-----
/u03/oradata/CURSOxy/temp01.dbf
```

```
SQL> select member from v$logfile;
MEMBER
```

```
-----
/u04/oradata/CURSOxy/redo01.log
/u04/oradata/CURSOxy/redo02.log
/u04/oradata/CURSOxy/redo03.log
```

```
SQL> select name from v$controlfile;
```

NAME

 /u02/oradata/CURSOxy/control11.ctl
 /u03/oradata/CURSOxy/control2.ctl

1.6. Identificar la estructura lógica de la BD: tablespaces, segmentos, extensiones.

En el Diccionario de Datos (DD) de la BD tenemos vistas para comprobar la estructura lógica de la BD: DBA_TABLESPACES, DBA_DATA_FILES, DBA_SEGMENTS y DBA_EXTENTS.

```
select * from dba_tablespaces;
select * from dba_datafiles;
select * from dba_tempfiles;
select * from dba_segments;
select * from dba_extents;
```

Solución:

(Para ver los tablespaces)
 SQL> select tablespace_name from dba_tablespaces
 order by tablespace_name;

TABLESPACE_NAME

 SYSAUX
 SYSTEM
 TEMP
 UNDO_RBS
 USERS

(Para ver los ficheros de datos)
 SQL> select tablespace_name, file_name from dba_data_files
 order by tablespace_name, file_name;

TABLESPACE_NAME

 FILE_NAME

 SYSAUX
 /u03/oradata/CURSOxy/sysaux01.dbf

SYSTEM
 /u02/oradata/CURSOxy/system01.dbf

UNDO_RBS
 /u03/oradata/CURSOxy/undo_rbs01.dbf

USERS
 /u02/oradata/CURSO22/users01.dbf

(Para ver los ficheros de los tablespaces temporales)
 SQL> select tablespace_name, file_name from dba_temp_files
 order by tablespace_name, file_name;

TABLESPACE_NAME

 FILE_NAME

 TEMP

Administración Básica de Oracle10g

/u03/oradata/CURSOxy/temp01.dbf

(Para ver cuántos segmentos hay en cada tablespace, agrupados por tipos de segmento)

```
SQL> select tablespace_name,segment_type,count(*) segmentos
       from dba_segments
       group by tablespace_name,segment_type;
```

TABLESPACE_NAME	SEGMENT_TYPE	SEGMENTOS
SYSTEM	CLUSTER	10
SYSTEM	INDEX	587
SYSTEM	TABLE	461
SYSAUX	LOBSEGMENT	43
SYSAUX	INDEX	235
SYSAUX	INDEX PARTITION	80
SYSAUX	LOB PARTITION	1
SYSAUX	NESTED TABLE	1
USERS	TABLE	4
SYSTEM	LOBSEGMENT	72
UNDO_RBS	TYPE2 UNDO	8
SYSAUX	LOBINDEX	43
SYSAUX	TABLE PARTITION	72
SYSAUX	TABLE	215
SYSTEM	NESTED TABLE	3
SYSTEM	ROLLBACK	1
SYSTEM	LOBINDEX	72
USERS	INDEX	2

18 filas seleccionadas.

(Para ver cuántas extensiones de cada tipo de segmento, hay en cada tablespace)

```
SQL> select tablespace_name,segment_type,count(*)
       from dba_extents
       group by tablespace_name,segment_type;
```

TABLESPACE_NAME	SEGMENT_TYPE	EXTENSIONES
SYSTEM	CLUSTER	66
SYSTEM	TABLE	748
SYSTEM	INDEX	837
SYSAUX	NESTED TABLE	1
USERS	TABLE	4
SYSAUX	INDEX	245
SYSAUX	INDEX PARTITION	91
SYSAUX	LOBSEGMENT	44
SYSAUX	LOB PARTITION	1
SYSTEM	LOBSEGMENT	109
UNDO_RBS	TYPE2 UNDO	37
SYSAUX	TABLE PARTITION	89
SYSAUX	LOBINDEX	43
SYSAUX	TABLE	252
SYSTEM	NESTED TABLE	3
SYSTEM	ROLLBACK	7
SYSTEM	LOBINDEX	72
USERS	INDEX	2

18 filas seleccionadas.

1.7. Consultar información sobre la base de datos (v\$database) y la instancia (v\$instance).

Podemos obtener información de la base de datos y de la instancia, de las vistas V\$DATABASE y V\$INSTANCE, respectivamente.

Solución:

```
SQL> select name, created, log_mode, checkpoint_change#, open_mode, platform_name,
current_scn from v$database;
```

```
NAME          CREATED LOG_MODE          CHECKPOINT_CHANGE# OPEN_MODE
-----
PLATFORM_NAME
-----
CURRENT_SCN
-----
CURSORoxy    09/08/06 NOARCHIVELOG          199194 READ WRITE
Linux IA (32-bit)
199233
```

```
SQL> select instance_name,host_name,version,startup_time,
status,archiver,logins,database_status from v$instance;
```

```
INSTANCE_NAME
-----
HOST_NAME
-----
VERSION          STARTUP_T STATUS          ARCHIVE LOGINS          DATABASE_STATUS
-----
CURSORoxy
cursos.atica.um.es
10.2.0.2.0          18/10/06 OPEN          STOPPED ALLOWED          ACTIVE
```

1.8. Localizar el proceso “servidor” asociado a mi sesión (v\$process y v\$session). ¿Es un servidor dedicado o compartido?

Toda sesión contra la BD tiene dos procesos asociados: cliente y servidor. En el cliente tenemos el proceso de usuario que inicia la sesión y en el servidor de base de datos tendremos el proceso que sirve las peticiones de dicha sesión; que puede ser un servidor dedicado o compartido. En las vistas V\$SESSION y V\$PROCESS tenemos toda la información relativa a sesiones y procesos, respectivamente.

Solución:

```
SQL> connect SYSTEM
```

```
SQL> select a.SERVER, a.username dbuser,a.OSUSER, a.PROCESS user_process,
a.machine, a.terminal, a.program user_program,
b.spid server_process, b.program server_program
from v$session a, v$process b
where a.username=USER and a.PADDR=b.ADDR;
```

```
SERVER          DBUSER          OSUSER
-----
USER_PROCESS MACHINE
-----
TERMINAL          USER_PROGRAM
-----
SERVER_PROCE SERVER_PROGRAM
-----
DEDICATED SYSTEM          cursoXY
30580          cursos.atica.um.es
pts/11          sqlplus@cursos.atica.um.es (TNS V1-V3)
30581          oracle@cursos.atica.um.es (TNS V1-V3)
```

```
SQL> !ps -fp 30581
```

Administración Básica de Oracle10g

```
UID          PID  PPID  C STIME TTY          TIME CMD
oracle    30581 30580  0 13:36 ?           00:00:01 oracleCURSOxy
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq))
```

En este caso, el proceso servidor asociado a mi sesión es un servidor DEDICADO.

1.9. ¿Cuanto ocupa la Dictionary cache y la Library cache en tu BD? (v\$sstat)

En la vista V\$SGASTAT hay información detallada sobre las partes de la SGA.

Solución:

```
SQL> select * from v$sgastat where name like '%cache';
POOL          NAME                                BYTES
-----
shared pool   buffer_cache                        41943040
shared pool   library cache                       2578224
shared pool   row cache                           3741868
```

1.10. Ver la actividad de la Library Cache (v\$llibrarycache).

En la vista V\$LIBRARYCACHE podemos ver los ratios de eficiencia de la Library Cache, en “tantos por uno”. El objetivo es q se aproximen los más posible a 1, de lo contrario es posible que haya que aumentar el tamaño de la Shared Pool.

Solución:

```
SQL> select namespace, pinhitratio from v$llibrarycache;
NAMESPACE          PINHITRATIO
-----
SQL AREA           ,878383629
TABLE/PROCEDURE    ,511005966
BODY               ,368421053
TRIGGER            1
INDEX              0
CLUSTER            ,956521739
OBJECT             1
PIPE               1
JAVA SOURCE        1
JAVA RESOURCE      1
JAVA DATA         1
11 rows selected.
```

1.11. Ver las sentencias SQL que guarda la Shared-Pool (v\$sqlarea).

En la vista V\$SQLAREA podemos ver el contenido del “área SQL” de la sharedpool, así como información útil para el ajuste de cada una de las sentencias sql (Shared Pool consumida, nº de veces q se ha ejecutado, nº de veces q se ha salido de la caché, lecturas físicas, tiempo de CPU, tiempo total incluyendo compilación, etc).

Solución:

```
SQL> SET PAUSE ON
SQL> SET PAGESIZE 37
SQL> select SQL_TEXT, PERSISTENT_MEM, EXECUTIONS, LOADS, DISK_READS, CPU_TIME,
ELAPSED_TIME from v$sqlarea order by DISK_READS desc;
SQL_TEXT
-----
PERSISTENT_MEM EXECUTIONS          LOADS DISK_READS    CPU_TIME ELAPSED_TIME
-----
select tablespace_name,segment_type,count(*) from dba_extents group by tablesp
ace_name,segment_type
      85108              1              1          1793      785150      6289684

select /*+ index(idl_ub2$ i_idl_ub21) */ piece#,length,piece from idl_ub2$ wher
e obj#=:1 and part=:2 and version=:3 order by piece#
      3516              42              1           293      75551      202647

select tablespace_name,segment_type,count(*) segmentos from dba_segments group
by tablespace_name,segment_type
      40548              1              1           219      127155      210013
...
```

1.12. Crear el fichero de autenticación y activarlo (orapwd).

Los pasos a seguir son:

- Crear fichero en \$ORACLE_HOME/dbs con la utilidad orapwd (si ya existe y queremos volver a crearlo, usaremos "force=y"). Si creamos el fichero nuevo desde un usuario diferente al que está ejecutando el sw de Oracle, hemos de tener en cuenta que, después de crear el fichero le tendremos que dar permiso de escritura al usuario que está ejecutando el sw de Oracle (p.e. desde linux con "chmod").
- Añadir el parámetro remote_login_passwordfile al init.ora.
- Parar y arrancar la BD.

Solución:

```
$ ls -l $ORACLE_HOME/dbs/orapw$ORACLE_SID
-rw-rw---- 1 oracle dba 2048 oct 18 13:01
/u01/app/oracle/product/10.2.0.1/dbs/orapwCURSOxy
$ orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID password=miclave entries=5 force=y
(La opción "force=y" es para machacarlo si ya existe)
```

(El siguiente paso sólo es necesario si el fichero no es propiedad del usuario q está ejecutando el sw Oracle, y dicho usuario no tuviera permiso de escritura sobre el fichero de claves creado)

```
$ chmod g+w $ORACLE_HOME/dbs/orapwCURSOxy
```

```
$ echo "remote_login_passwordfile=EXCLUSIVE" >>
/u01/app/oracle/admin/CURSOxy/pfile/initCURSOxy.ora
$ echo >> /u01/app/oracle/admin/CURSOxy/pfile/initCURSOxy.ora
```

```
SQL> CONNECT / AS SYSDBA
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
...
SQL> exit
```

```
SQL> CONNECT SYS@CURSOxy AS SYSDBA
Enter password: miclave
SQL> EXIT
```

1.13. Asignar la variable NLS_LANG para nuestro país y comprobar el cambio en las respuestas de Oracle desde sqlplus.

- Comprobar el valor de NLS_LANG y ver la fecha del sistema desde sqlplus.
- Salir de sqlplus y asignar spanish_spain a NLS_LANG.
- Ejecutar sqlplus y comprobar que pide “usuario” y no “username”.
- Comprobar de nuevo la fecha del sistema desde sqlplus y verificar el cambio de formato.

Solución:

```
$ echo $NLS_LANG
spanish_spain
```

```
$ sqlplus
Introduzca el nombre de usuario: / as sysdba
SQL> exit
```

```
$ export NLS_LANG=american_america
$ sqlplus
Enter user-name: / as sysdba
SQL> exit
```

```
$ export NLS_LANG=spanish_spain
```

1.14. Subir el tamaño de la shared-pool un gránulo más (p.e. si tenía 48M subirlo a 52M, si el gránulo es de 4M) y comprobar cómo aumenta el espacio libre en dicha caché.

- Comprobar el valor de shared_pool_size, así como el espacio libre en la shared_pool.
- Asignarle 52M y volver a comprobar el valor del parámetro, así como el espacio libre que tiene ahora la shared-pool.
- Finalmente, volver a dejar la shared-pool como estaba inicialmente (0) y comprobar de nuevo los valores anteriores.

Solución:

(shared_pool_size vale cero pq se está usando la gestión automática de la SGA de Oracle 10g; de modo q se irá ajustando el tamaño de la Shared Pool automáticamente según las necesidades)

```
SQL> show parameter shared_pool_size
```

NAME	TYPE	VALUE
shared_pool_size	big integer	0

```
SQL> select * from v$sgainfo;
```

NAME	BYTES	RES
...		
Buffer Cache Size	41943040	Yes

Administración Básica de Oracle10g

```

Shared Pool Size          50331648 Yes
...
Granule Size              4194304 No
Maximum SGA Size         83886080 No
...
Free SGA Memory Available 20971520
  
```

```

SQL> select * from v$sga_dynamic_components where component='shared pool';
COMPONENT                                CURRENT_SIZE
-----
MIN_SIZE  MAX_SIZE  USER_SPECIFIED_SIZE  OPER_COUNT  LAST_OPER_TYP  LAST_OPER
LAST_OPE GRANULE_SIZE
-----
shared pool                                50331648
 29360128          0                0           3 GROW          IMMEDIATE
27/02/07          4194304
  
```

```

SQL> select sum(bytes) from v$sgastat where pool='shared pool' and name like '%free
%';
SUM(BYTES)
-----
 11108980
  
```

```

SQL> alter system set shared_pool_size=52M;
System altered.
  
```

```

SQL> show parameter shared_pool_size
NAME                                TYPE                                VALUE
-----
shared_pool_size                    big integer                          52M
  
```

```

SQL> select * from v$sgainfo;
NAME                                BYTES  RES
-----
...
Buffer Cache Size                   37748736 Yes
Shared Pool Size                     54525952 Yes
...
Granule Size                         4194304 No
...
Free SGA Memory Available            20971520
  
```

```

SQL> select * from v$sga_dynamic_components where component='shared pool';
COMPONENT                                CURRENT_SIZE
-----
MIN_SIZE  MAX_SIZE  USER_SPECIFIED_SIZE  OPER_COUNT  LAST_OPER_TYP  LAST_OPER
LAST_OPE GRANULE_SIZE
-----
shared pool                                54525952
 29360128          0                54525952       4 GROW          MANUAL
27/02/07          4194304
  
```

```

SQL> select sum(bytes) from v$sgastat where pool='shared pool' and name like '%free
%';
SUM(BYTES)
-----
 15214916
  
```

```

SQL> alter system set shared_pool_size=0;
System altered.
  
```

Administración Básica de Oracle10g

```
SQL> show parameter shared_pool_size
NAME                                TYPE                                VALUE
-----                                -                                -
shared_pool_size                    big integer                        0

SQL> select * from v$sga_dynamic_components where component='shared pool';
COMPONENT                                CURRENT_SIZE
-----                                -
  MIN_SIZE  MAX_SIZE USER_SPECIFIED_SIZE OPER_COUNT LAST_OPER_TYP LAST_OPER
-----
LAST_OPE GRANULE_SIZE
-----
shared pool                                54525952
  29360128          0          0          4 GROW          MANUAL
27/02/07          4194304
```

1.15. Comprobar el funcionamiento de la caché de redolog, como protectora del contenido de la caché de datos. Para ello iniciaremos una transacción y provocaremos una caída de la BD, comprobando que al arrancarla de nuevo, se mantendrá la integridad de la misma.

- Crear la tabla BORRAME del usuario SCOTT.
- Insertar una fila sin hacer commit y forzar la caída de la BD.
- Arrancar de nuevo la BD y comprobar que la fila insertada no está (pues no se hizo commit).
- Repetir la inserción de la fila, esta vez haciendo commit; y forzar la caída de la BD otra vez.
- Arrancar la BD una vez más y comprobar que ahora la fila sí está (ya que se validó la transacción con commit).

Solución:

```
SQL> connect / as sysdba
Connected.

SQL> create table SCOTT.borraime (c1 varchar2(10)) tablespace users;
Table created.

SQL> desc SCOTT.borraime
Name                                Null?                                Type
-----                                -                                -
C1                                    VARCHA2(10)

SQL> insert into SCOTT.borraime values ('Primera');
1 row created.

SQL> commit;
Commit complete.

SQL> select * from SCOTT.borraime;
C1
-----
Primera

SQL> insert into SCOTT.borraime values ('Segunda');
1 row created.

SQL> select * from SCOTT.borraime;
C1
-----
```

Primera
Segunda

```
SQL> shutdown abort
ORACLE instance shut down.
```

```
SQL> connect / as sysdba
Connected to an idle instance.
```

```
SQL> startup
...
Database opened.
```

```
SQL> select * from SCOTT.borrarme;
C1
```

```
-----
Primera
```

```
SQL> insert into SCOTT.borrarme values ('Segunda');
1 row created.
```

```
SQL> commit;
Commit complete.
```

```
SQL> select * from SCOTT.borrarme;
C1
```

```
-----
Primera
Segunda
```

```
SQL> shutdown abort
ORACLE instance shut down.
```

```
SQL> connect / as sysdba
Connected to an idle instance.
```

```
SQL> startup
...
Database opened.
```

```
SQL> select * from SCOTT.borrarme;
C1
```

```
-----
Primera
Segunda
```

1.16. Comprobar el funcionamiento de la caché de datos, en lo que se refiere a la mejora del rendimiento cuando se repite una consulta. ¿Por qué la segunda vez que se lanza la misma consulta tarda menos?

- Activar la medición de tiempos en sqlplus con SET TIMING ON.
- Lanzar la consulta SELECT COUNT(*) FROM DBA_SOURCE.
- Volver a lanzar la misma consulta.
- Comprobar que la segunda ejecución tarda mucho menos, ya que los datos ya se cargaron en la caché de datos al lanzarla la primera vez; y por tanto se acceden directamente en memoria y no en disco.

Solución:

(Primero vamos a vaciar la caché de datos (buffer cache), para asegurarnos que los datos no estén en ella)

```
SQL> alter system flush buffer_cache;  
Sistema modificado.
```

```
SQL> set timing on
```

```
SQL> select count(*) from dba_source;  
COUNT(*)
```

```
-----  
101914  
Elapsed: 00:00:01.04
```

```
SQL> r
```

```
COUNT(*)
```

```
-----  
101914  
Elapsed: 00:00:00.03
```