

## **PRACTICAS TEMA 2.**

### **DISEÑO Y CREACION DE UNA BASE DE DATOS.**

2.1. Diseñar, sobre el papel, una BD teniendo en cuenta las contenedores de E/S y la organización de los tablespaces básicos. Suponer 4 discos: /u01 (SW) y /u02, /u03, /u04 (BD).

2.2. Crear un init.ora, diseñando requerimientos de memoria, suponiendo 1Gb de RAM y 21 instancias con 10 usuarios cada una. Se puede usar un máximo del 75% de la RAM para las SGAs de todas las bases de datos. Los checkpoints sucederán cuando se llene el fichero redolog y, como máximo, cada 30 minutos. La gestión de rollback será automática. Tendremos dos copias del fichero de control en /u02 y /u03.

2.3. Crear la BD que hemos diseñado anteriormente, y para la que también hemos creado un init.ora. Hacerlo manualmente con CREATE DATABASE. El nombre de la BD será ALUxy. Antes de lanzar el comando de creación, ponte de acuerdo con un compañero para que no lo lancéis a la vez.

2.4. Crear la misma BD anterior, esta vez usando OMF, y cambiando el nombre por OMFxy, creando los ficheros bajo /u02/oradata/OMFxy y usando dos destinos para redolog (en /u03 y /u04).

2.5. Consultar el "Simulador de DBCA".

#### Recursos.

Máquina: [cursos.atca.um.es](http://cursos.atca.um.es)

Usuario unix: curso51 a curso70

Conexión máquina unix: ssh con Secure Shell

Bases de datos: CURSO51 a CURSO70

Conexión a bd: sqlplus desde sesión unix, usuario "/" as sysdba"

## 2.1. Diseñar, sobre el papel, una BD teniendo en cuenta las contenedores de E/S y la organización de los tablespaces básicos. Suponer 4 discos: /u01 (SW) y /u02, /u03, /u04 (BD).

- Comprobar discos reales del servidor y ocupación (df). También actividad (iostat). El SW de Oracle ya está instalado en el disco sda (/ , concretamente /u01) y la BD en el disco sdb (/home, concretamente /u02, /u03 y /u04 (los dos primeros son enlaces a /home/u02 y /home/u03, respectivamente). En este apartado, vamos a suponer que no hay nada instalado, y en función de los datos que nos da "df" e "iostat", vamos a hacer un supuesto reparto del SW y BD, de forma que supondremos que el SW necesitaría 3Gb y la BD 4Gb.
- Ahora vamos a diseñar una base de datos ALUxy, para la que disponemos de tres directorios (imaginemos que son discos): /u02, /u03 y /u04). Además de los tablespaces básicos (system, temp y undo\_rbs) vamos a añadir los tablespaces: tools, users, datos, índices, estatico\_dat y estatico\_ind.
- Tablespaces system (/u03), temp (/u03), undo\_rbs (/u03), tools (/u02), users (/u02), datos (/u02), índices (/u03), estáticos (/u02), etc.

### Solución:

- Comprobar espacio en disco y actividad de los mismos, y decidir como repartir SW (3Gb) y BD(4Gb).

```
$ df -m
S.ficheros          Bloques de 1M  Usado  Dispon  Uso%  Montado en
/dev/sda2           50396      14226   33611   30%  /
/dev/sda1             99          9        86     9%  /boot
none                 506         0        506    0%  /dev/shm
/dev/sdb2           100792     28774   66899   31%  /home
```

```
$ iostat 5 5
...
Device:            tps    Blq_leid/s    Blq_escr/s    Blq_leid    Blq_escr
sda                 1,39         7,29         26,25    10216884    36772686
sdb                 7,97        22,50        91,80    31519526    128627672
...
Device:            tps    Blq_leid/s    Blq_escr/s    Blq_leid    Blq_escr
sda                 0,40         0,00         8,00         0           40
sdb                11,60         0,00        123,20         0          616
```

Comprobamos que la máquina tiene dos discos (sda y sdb). En /home (/dev/sdb2) tenemos unos 60Gb libres, y en / (/dev/sda2) tenemos uno 30Gb. Además el disco sda tiene menos E/S que sdb. Vamos a separar el software de la BD, dedicando la partición más pequeña para el SW (/) y la más grande para la BD (/home). De esta forma, **definiríamos los directorios /u01 para el SW y /home/u02 (creando el enlace /u02) para la BD.** Además, en cuanto a la BD se refiere, separaremos el redo de los datos, creando /u04 para los ficheros redo log.

- Diseñar, sobre el papel, la BD ALUxy.

Vamos a separar datos, índices y redo; en /u02, /u03 y /u04, respectivamente. Además, el tablespace SYSTEM (que contiene el DD) lo vamos a ubicar con los índices (/u03) y TEMP y UNDO\_RBS con los datos (/u02). Estamos suponiendo que la mayor actividad de E/S se va a presentar en datos, SYSTEM y redo, pero podría no ser así; de hecho una vez creada la BD deberemos revisar el estado de la E/S (iostat).

Resumiendo, los tablespaces de ALUxy quedaría repartidos, por ejemplo, así:

```
/u02/oradata/ALUxy -> DATOS, ESTATICO_DAT, TOOLS, USERS,
                        TEMP, UNDO_RBS
/u03/oradata/ALUxy -> INDICES, ESTATICO_IND, SYSTEM
/u04/oradata/ALUxy -> Redo
```

**2.2. Crear un init.ora, diseñando requerimientos de memoria, suponiendo 1Gb de RAM y 21 instancias con 10 usuarios cada una. Se puede usar un máximo del 75% de la RAM para las SGAs de todas las bases de datos. Los checkpoints sucederán cuando se llene el fichero redolog y, como máximo, cada 30 minutos. La gestión de rollback será automática. Tendremos dos copias del fichero de control en /u02 y /u03.**

### Solución:

Comprobamos la memoria y su estado con el comando "free -m":

```
$ free -m
      total        used         free       shared    buffers     cached
Mem:      1010         982           28           0         260         378
-/+ buffers/cache:
Swap:    2047           2        2044
```

El 75% de 1010M son 757M y si lo dividimos entre 21, nos quedan 36M máximos por cada SGA; así que asignaremos sga\_max\_size=36M. De estos vamos a destinar el 45% a shared\_pool y otro 45% a db\_cache\_size. El 45% de 36M son 16M para cada uno.

```
$ bc
...
1010*.75
757.50
757/21
36
36*.45
16.20
quit
```

Ya tenemos 32M de la SGA asignados: shared\_pool\_size=16M, db\_cache\_size=16M. Como no vamos a usar ni Java ni MTS, haremos java\_pool\_size=0 y large\_pool\_size=0. Nos queda log\_buffer, que le podemos asignar 1M (1048576). Resumiendo, en cuanto a asignación de memoria se refiere:

```
sga_max_size=36M
shared_pool_size=16M
db_cache_size=16M
log_buffer=1048576
large_pool=0
java_pool=0
```

Por otro lado, partimos del supuesto de que el número de sesiones simultáneas será 10; por tanto fijaremos processes=10 + nºprocesos\_background +1. Los procesos background mínimos son seis (DBWR, LGWR, SMON, PMON, CKPT y RECO); y sumamos uno por el proceso PSEUDO (utilizado para la transición entre procesos); en total tenemos:

```
processes=17
```

Como queremos rollback automático, y si llamamos al tablespace de undo UNDO\_RBS, asignamos los parámetros:

```
undo_management=auto  
undo_tablespace=undo_rbs
```

El nombre de la base de datos es ALUxy, donde xy es el número que incluye tu nombre de usuario linux (por ejemplo, para curso30 sería ALU30):

```
db_name=ALUxy
```

Como tenemos instalado Oracle9i (9.2.0.1), y queremos utilizar todas sus novedades, añadimos:

```
compatible=9.2.0
```

Para tener dos copias del fichero de control en /u02 y /u03, asignamos:

```
control_files=(/u02/oradata/ALUxy/control1.ctl, /u03/oradata/ALUxy/control2.ctl)
```

Y, finalmente, las trazas las vamos a generar en /u01/app/oracle/admin/ALUxy, en los directorios bdump, cdump y udump; para el alert, cores y trazas de usuario, respectivamente, limitando el tamaño del fichero a 5Mb (en bloques del SO, 10240):

```
background_dump_dest=/u01/app/oracle/admin/ALUxy/bdump  
core_dump_dest=/u01/app/oracle/admin/ALUxy/cdump  
user_dump_dest=/u01/app/oracle/admin/ALUxy/udump  
max_dump_file_size=10240
```

De esta forma el init podría quedar así:

```
db_name=ALUxy  
compatible = 9.2.0  
control_files = (/u02/oradata/ALUxy/control1.ctl, /u03/oradata/ALUxy/control2.ctl)  
undo_management = auto  
undo_tablespace = undo_rbs  
background_dump_dest = /u01/app/oracle/admin/ALUxy/bdump  
core_dump_dest = /u01/app/oracle/admin/ALUxy/cdump  
user_dump_dest = /u01/app/oracle/admin/ALUxy/udump  
max_dump_file_size = 10240  
db_cache_size = 16M  
shared_pool_size = 16M  
large_pool_size = 0  
java_pool_size = 0  
log_buffer = 1048576  
log_checkpoint_interval = 0  
log_checkpoint_timeout = 1800  
sga_max_size = 36M  
processes = 17
```

**Lo guardamos como initALUxy.ora, en el directorio \$ORACLE\_HOME/dbs.**

**2.3. Crear la BD que hemos diseñado anteriormente, y para la que también hemos creado un init.ora. Hacerlo manualmente con CREATE DATABASE. El nombre de la BD será ALUxy. Antes de lanzar el comando de creación, ponte de acuerdo con un compañero para que no lo lanceís a la vez.**

### Solución:

Con el comando CREATE DATABASE vamos a crear una BD con los tablespaces básicos: SYSTEM, TEMP y UNDO\_RBS; más dos copias del fichero de control y tres ficheros redolog (tres grupos de un sólo miembro cada uno).

Según el diseño anterior, la distribución de ficheros sería la siguiente:

```
/u02/oradata/ALUxy -> TEMP, UNDO_RBS, control1
/u03/oradata/ALUxy -> SYSTEM, control2
/u04/oradata/ALUxy -> Redo
```

Debemos tener el fichero init.ora anteriormente creado en:

**\$ORACLE\_HOME/dbs/initALUxy.ora**

El comando CREATE DATABASE podría quedar así:

```
CREATE DATABASE "ALUxy"
  maxdatafiles 254
  maxinstances 1
  maxlogfiles 32
  character set WE8ISO8859P15
DATAFILE '/u02/oradata/ALUxy/system01.dbf' SIZE 260M
  AUTOEXTEND ON NEXT 10M MAXSIZE 400M
  EXTENT MANAGEMENT LOCAL
UNDO TABLESPACE UNDO_RBS
  DATAFILE '/u03/oradata/ALUxy/undo_rbs01.dbf' SIZE 10M
  AUTOEXTEND ON NEXT 1M MAXSIZE 20M
DEFAULT TEMPORARY TABLESPACE TEMP
  TEMPFILE '/u03/oradata/ALUxy/temp01.dbf' SIZE 10M REUSE
  AUTOEXTEND ON NEXT 10M MAXSIZE 50M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K
logfile '/u04/oradata/ALUxy/redo01.log' SIZE 10M,
  '/u04/oradata/ALUxy/redo02.log' SIZE 10M,
  '/u04/oradata/ALUxy/redo03.log' SIZE 10M;
```

Asignamos la variable de entorno ORACLE\_SID=ALUxy, para que Oracle sepa el nombre de la instancia que vamos a crear:

```
/home/CURSO/curso01 (CURSOxy)> export ORACLE_SID=ALUxy
/home/CURSO/curso01 (ALUxy)>
```

Antes de crear la BD debemos crear los directorios necesarios:

```
$ mkdir /u02/oradata/ALUxy
$ mkdir /u03/oradata/ALUxy
$ mkdir /u04/oradata/ALUxy
$ mkdir /u01/app/oracle/admin/ALUxy
$ mkdir /u01/app/oracle/admin/ALUxy/bdump
```

## Administración Avanzada de Oracle9i

```
$ mkdir /u01/app/oracle/admin/ALUxy/cdump
$ mkdir /u01/app/oracle/admin/ALUxy/udump
```

Antes de crear la BD, debemos dar permiso de escritura al usuario oracle sobre los directorios que hemos creado. Lo haremos dando permiso de escritura al grupo:

```
$ chmod g+w /u0?/oradata/ALUxy
$ chmod -R g+w /u01/app/oracle/admin/ALUxy
```

Ahora sí que podemos crear la BD. **Pregunta a tu compañero si ha iniciado el proceso de creación, de forma que si lo ha hecho debes esperarte a que el suyo termine:**

```
/home/CURSO/curso01 (ALU22)> sqlplus
...
Enter user-name: / as sysdba
Connected to an idle instance.
SQL> startup nomount
ORACLE instance started.
Total System Global Area      47779900 bytes
Fixed Size                     450620 bytes
Variable Size                  29360128 bytes
Database Buffers               16777216 bytes
Redo Buffers                    1191936 bytes

SQL> CREATE DATABASE "ALUxy"
 2      maxdatafiles 254
 3      maxinstances 1
 4      maxlogfiles 32
 5      character set WE8ISO8859P15
 6  DATAFILE '/u02/oradata/ALUxy/system01.dbf' SIZE 260M
 7      AUTOEXTEND ON NEXT 10M MAXSIZE 400M
 8      EXTENT MANAGEMENT LOCAL
 9  UNDO TABLESPACE UNDO_RBS
10      DATAFILE '/u03/oradata/ALUxy/undo_rbs01.dbf' SIZE 10M
11      AUTOEXTEND ON NEXT 1M MAXSIZE 20M
12  DEFAULT TEMPORARY TABLESPACE TEMP
13      TEMPFILE '/u03/oradata/ALUxy/temp01.dbf' SIZE 10M REUSE
14      AUTOEXTEND ON NEXT 10M MAXSIZE 50M
15      EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K
16  logfile '/u04/oradata/ALUxy/redo01.log' SIZE 10M,
17      '/u04/oradata/ALUxy/redo02.log' SIZE 10M,
18*     '/u04/oradata/ALUxy/redo03.log' SIZE 10M;
Database created.
```

Vamos a comprobar los ficheros que componen las BD que hemos creado:

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

```
-----
/u02/oradata/ALU22/system01.dbf
/u03/oradata/ALU22/undo_rbs01.dbf
```

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

```
-----
/u03/oradata/ALU22/temp01.dbf
```

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

```
-----
/u04/oradata/ALU22/redo01.log
/u04/oradata/ALU22/redo02.log
/u04/oradata/ALU22/redo03.log
```

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

```
-----
/u02/oradata/ALU22/control1.ctl
/u03/oradata/ALU22/control2.ctl
```

Finalmente, cerramos la BD:

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

Las vistas del DD no las vamos a crear porque tardan unos 15 minutos, y habría que hacerlo uno por uno (alumno por alumno). Como sois 20 alumnos, necesitaríamos 300 minutos (5 horas) para completarlo.

## **2.4. Crear la misma BD anterior, esta vez usando OMF, y cambiando el nombre por OMFxy, creando los ficheros bajo /u02/oradata/OMFxy y usando dos destinos para redolog (en /u03 y /u04).**

### **Solución:**

En primer lugar, creamos el init.ora, copia del anterior, pero cambiando ALU por OMF:

```
$ORACLE_HOME/dbs/initOMFxy.ora
```

Tenemos que cambiar algunas cosas, fundamentalmente añadir los parámetros de activación de OMF: db\_create\_file\_dest, db\_create\_online\_log\_dest\_1 y db\_create\_online\_log\_dest\_2:

```
db_name=OMFxy
compatible = 9.2.0
control_files = (/u02/oradata/OMFxy/control1.ctl,
/u03/oradata/OMFxy/control2.ctl)undo_management = auto
undo_tablespace = undo_rbs
DB_CREATE_FILE_DEST=/u02/oradata/OMFxy
DB_CREATE_ONLINE_LOG_DEST_1=/u03/oradata/OMFxy
DB_CREATE_ONLINE_LOG_DEST_2=/u04/oradata/OMFxy
background_dump_dest = /u01/app/oracle/admin/OMFxy/bdump
core_dump_dest = /u01/app/oracle/admin/OMFxy/cdump
user_dump_dest = /u01/app/oracle/admin/OMFxy/udump
max_dump_file_size = 10240
db_cache_size = 16M
shared_pool_size = 16M
large_pool_size = 0
java_pool_size = 0
log_buffer = 1048576
log_checkpoint_interval = 0
log_checkpoint_timeout = 1800
sga_max_size = 36M
processes = 17
```

## Cambiamos la variable ORACLE\_SID=OMFxy:

```
$ export ORACLE_SID=OMFxy
/home/CURSO/curso01 (OMFxy)>
```

## Vamos a crear los directorios necesarios para los ficheros de la BD:

```
/home/CURSO/curso01 (OMFxy)> mkdir /u02/oradata/OMFxy
/home/CURSO/curso01 (OMFxy)> mkdir /u03/oradata/OMFxy
/home/CURSO/curso01 (OMFxy)> mkdir /u04/oradata/OMFxy
/home/CURSO/curso01 (OMFxy)> mkdir /u01/app/oracle/admin/OMFxy
/home/CURSO/curso01 (OMFxy)> mkdir /u01/app/oracle/admin/OMFxy/bdump
/home/CURSO/curso01 (OMFxy)> mkdir /u01/app/oracle/admin/OMFxy/cdump
/home/CURSO/curso01 (OMFxy)> mkdir /u01/app/oracle/admin/OMFxy/udump
```

Antes de crear la BD, debemos dar permiso de escritura al usuario oracle sobre los directorios que hemos creado. Lo haremos dando permiso de escritura al grupo:

```
$ chmod g+w /u02/oradata/OMFxy
$ chmod -R g+w /u01/app/oracle/admin/OMFxy
```

Ahora construimos la sentencia de creación de la BD, que es muy simple al usar OMF:

```
CREATE DATABASE OMFxy
DEFAULT TEMPORARY TABLESPACE TEMP
UNDO TABLESPACE UNDO_RBS;
```

Vamos a crear la BD. Antes, igual que en el caso anterior, ponte **de acuerdo con tu compañero de forma que no lancéis el comando de creación a la vez.**

```
home/CURSO/curso01 (OMF22)> sqlplus
```

```
...
Enter user-name: / as sysdba
Connected to an idle instance.
SQL> startup nomount
ORACLE instance started.
Total System Global Area 47779900 bytes
Fixed Size 450620 bytes
Variable Size 29360128 bytes
Database Buffers 16777216 bytes
Redo Buffers 1191936 bytes
```

```
SQL> show parameter db_create
```

NAME	TYPE	VALUE
db_create_file_dest	string	/u02/oradata/OMF22
db_create_online_log_dest_1	string	/u03/oradata/OMF22
db_create_online_log_dest_2	string	/u04/oradata/OMF22

```
SQL> CREATE DATABASE "OMFxy"
2 DEFAULT TEMPORARY TABLESPACE TEMP
3* UNDO TABLESPACE UNDO_RBS;
Database created.
```

```
SQL> select name,bytes from v$datafile;
```

NAME	BYTES
/u02/oradata/OMF22/o1_mf_system_15wdp0hz_.dbf	104857600
/u02/oradata/OMF22/o1_mf_undo_rbs_15wdp9tq_.dbf	10485760

```
SQL> select name,bytes from v$tempfile;
```

NAME	BYTES
/u02/oradata/OMF22/o1_mf_temp_15wdpbnj_.tmp	104857600

```
SQL> SELECT name FROM V$CONTROLFILE;  
NAME
```

```
-----  
/u03/oradata/OMF22/o1_mf_15wdoqp3_.ctl  
/u04/oradata/OMF22/o1_mf_15wdoqpk_.ctl
```

```
SQL> select a.member, b.bytes from v$logfile a, v$log b where a.group#=b.group#;  
MEMBER                                BYTES
```

```
-----  
/u03/oradata/OMF22/o1_mf_1_15wdoqq8_.log 104857600  
/u04/oradata/OMF22/o1_mf_1_15wdorcc_.log 104857600  
/u03/oradata/OMF22/o1_mf_2_15wdotxy_.log 104857600  
/u04/oradata/OMF22/o1_mf_2_15wdox0k_.log 104857600
```

Finalmente cerramos la BD, como en el ejercicio anterior, con "shutdown immediate":

```
SQL> shutdown immediate  
Database closed.  
Database dismounted.  
ORACLE instance shut down.
```

## 2.5. Consultar el "Simulador de DBCA".

Se trata de un documento en PDF con las capturas de pantalla de la ejecución de una sesión del DataBase Configuration Assistant (dbca):

<http://cursos.atica.um.es/DBA9i2/pdf/contenidos/SimuladorDBCA.pdf>