

Práctica 4

Diseño de la estructura de datos de una empresa y
esquemas de copia de seguridad

Administración de bases de datos



**UNIVERSIDAD
DE GRANADA**

Grado en Ingeniería Informática - Sistemas de Información

Miguel Acedo Cañavate
Amador Carmona Méndez
Adrián Carmona Lupiáñez
Pablo M. Moreno Mancebo

Índice

1. Introducción	2
2. Estructura de almacenamiento	2
3. Ubicación Control File	3
4. Estructura de Tablespaces y Datafiles	5
5. Estructura de Redo Log	6
6. Estructura de Roles	7
7. Copia de Seguridad	9
8. Referencias Bibliográficas	10

1. Introducción

Nuestra entidad es una empresa destinada a la consultoría de recursos humanos compuesta por 100 trabajadores que se clasifican internamente de la siguiente manera.

El ápice estratégico compuesto por el nivel ejecutivo de la empresa la componen 5 personas. Los puestos de trabajo relacionados con el nivel gerencial de la empresa son ocupados por 15 personas. Por último, el nivel operativo de la empresa está compuesto por 85 trabajadores de los cuales 10 son trabajadores dedicados al desarrollo de aplicaciones que son usadas por el resto de los trabajadores del nivel operativo.

Este grupo de 10 personas, no solo se dedica al desarrollo de aplicaciones, sino que parte de ellos también son los encargados del mantenimiento del sistema.

2. Estructura de almacenamiento

Para el diseño del sistema necesario de nuestra empresa, la estructura de discos que hemos escogido está compuesta por siete discos en total y otra estructura de almacenamiento aparte de cintas magnéticas para los backups. Los tres primeros los dedicamos para la multiplexación del control file y almacenar los redo log files. El tamaño total de todos los redo log files asciende a 26 GB. La estructura de los discos uno, dos y tres dispone de espacio de sobra para si quisiéramos activar el modo archivado pudiéramos. Es algo que se hablará más adelante en el punto 4 de esta práctica.

Con el fin de no tener los cabezales de los disco moviéndose continuamente para escribir los redo y los data files, nuestra estructura diferencia los discos de redo log files y los de data files.

La estructura de los discos que almacenan los datafiles se justifica queriendo que no se satura ningún disco y que los demás estén ociosos. Con la estructura escogida, los cuatro discos se escribirán simultáneamente distribuyendo la carga de escritura entre ellos.

De la manera que hemos pensado la estructura de nuestro sistema, será necesario disponer de dos controladoras para cada uno de los grupos de discos que hemos considerado, en primer lugar tendremos una que se encargará de los redolog y control files y en segundo lugar la que se encargará de los datafiles.

DISCO 1 (256 GB SSD)	DISCO 2 (256 GB SSD)	DISCO 3 (256 GB SSD)
Control file	Control file	Control file
Redo log Files	Redo log Files	Redo log Files

DISCO 4 (4TB HDD)	DISCO 5 (4TB HDD)	DISCO 6 (4TB HDD)	DISCO 7 (4TB HDD)
2 datafile Explotación	2 datafile Explotación	2 datafile Explotación	2 datafile Explotación
2 datafile desarrollo	2 datafile desarrollo	1 datafile desarrollo	1 datafile desarrollo
1 datafile pruebas	1 datafile pruebas	1 datafile pruebas	1 datafile pruebas

Para las copias de seguridad:

CINTA MAGNÉTICA (15TB)	CINTA MAGNÉTICA (15TB)	CINTA MAGNÉTICA (15TB)	CINTA MAGNÉTICA (30TB)
Backup en caliente			Backup en frío

3. Ubicación Control File

Nuestro sistema hará uso de 3 control files, ya que para la estructura de redolog utilizaremos 3 discos, y estos se ubicaran juntos (control files y redo log file), para así poder minimizar el movimiento del cabezal y que sea más sencilla su lectura.

Para saber la ubicación de los control files si estamos dentro de la base de datos podemos saberlo con la sentencia:

```
SELECT value FROM V$parameter
WHERE name='control_files';
```

También podemos ubicar estas copias accediendo a la tabla controlfile mediante la sentencia

```
SELECT name FROM v$controlfile;
```

y nos daría la ruta de las copias del control file que haya en el sistema de la base de datos y sus nombres, como podemos observar en la siguiente imagen.

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

NAME
-----
/databases/oradata/ORADBA/control01.ctl
/databases/oradata/ORADBA/control02.ctl

SQL> select name from v$controlfile
2 ;

NAME
-----
/databases/oradata/ORADBA/control01.ctl
/databases/oradata/ORADBA/control02.ctl
```

Si la base de datos no está montada, podemos obtener la ubicación mediante el fichero de parámetros (pfile) accediendo a su ubicación y haciendo un **LESS init.ora** o un **cat int.ora**, siendo init.ora el fichero de parámetros, y al ejecutar esta sentencia, nos dará el contenido del pfile, y veremos una variable en pantalla llamada controlfile y la ruta de donde estos se encuentran como en la siguiente imagen.

```
*.compatible='19.0.0'
*.control_files='/databases/oradata/ORADBA/control01.ctl','/databases/oradata/ORADBA/control02.ctl'
*.db_block_size=8192
*.db_domain='pclab.localdomain'
*.db_name='oradba'
*.diagnostic_dest='/databases/app/oracle'
```

Estas dos imágenes son meramente de ejemplo para ilustrar donde podríamos ver las rutas de los control_files

Si quisiéramos aumentar el número de control files, porque queramos más redundancia, hayamos incluido un disco nuevo o cualquier otra razón tendremos que multiplexarlo. Como no se puede hacer cuando está la base de datos operativa, por ello hay que detener la instancia. Después copiamos el fichero de control a la nueva ubicación, se cambia el parámetro control_files en el archivo de parámetros (pfile). A continuación se levanta la base de datos con este archivo de parámetros, no confundir con el spfile que seguiría teniendo el enlace al anterior controlfile. comprobamos que todo funciona, comprobamos que haya un fichero mas de control, mediante las sentencias comentadas anteriormente, una vez que comprobamos que está el nuevo fichero creamos el SPFile a partir del que hemos utilizado para arrancar detenemos la BD y volvemos a arrancar pero esta vez con el SPFile, es decir hacemos un start up.

4. Estructura de Tablespaces y Datafiles

La estructura de tablespaces que vamos a utilizar en nuestra BD corresponderá a los distintos espacios de trabajo indicados en la práctica. Tendremos los siguientes tablespaces junto a sus tamaños máximos.

- Aplicaciones en desarrollo: 500GB
- Aplicaciones en pruebas: 110 GB
- Aplicaciones en explotación: 7,92 TB

El tamaño de los tablespaces y el número de extensiones mínimas y máximas, así como el tamaño de las extensiones se debe de pensar en base a la información proporcionada en el enunciado.

```
CREATE TABLESPACE desarrollo
DATAFILE  '/DISK4/dev01.dbf' SIZE 5G,
          '/DISK4/dev02.dbf' SIZE 5G,
          .
          .
          '/DISK7/dev06.dbf' SIZE 5G,
MINIMUM EXTENT 5G
DEFAULT STORAGE (INITIAL 5G NEXT 5G MAXEXTENTS 10);
```

```
CREATE TABLESPACE pruebas
DATAFILE  '/DISK4/test01.dbf' SIZE 5G,
          '/DISK5/test02.dbf' SIZE 5G,
          .
          .
          '/DISK7/test04.dbf' SIZE 5G,
MINIMUM EXTENT 5G
DEFAULT STORAGE (INITIAL 5G NEXT 5G MAXEXTENTS 5);
```

```
CREATE TABLESPACE explotación
DATAFILE  '/DISK4/app01.dbf' SIZE 10G,
          '/DISK4/app02.dbf' SIZE 10G
          .
          .
          '/DISK7/app08.dbf' SIZE 10G
MINIMUM EXTENT 10G
DEFAULT STORAGE (INITIAL 10G NEXT 20G MAXEXTENTS 50);
```

5. Estructura de Redo Log

Con las intenciones de usar una estructura circular, hemos decidido mantener el modo **archivado desactivado**. Aunque si fuese necesario activarlo, **se dispone de espacio suficiente** en los discos de redo log files para mantener el archivo durante cierto tiempo.

El tamaño de los redo log files lo hemos fijado en **1 GB** para que sea suficientemente grande para tener un número no excesivo de grupos y evitar por otro lado la pérdida de gran espacio cuando se realicen logs switches. Para el volumen de operaciones semanales del sistema que ascienden a 488000, queriendo que nuestra ventana de log sea de una semana y teniendo este número de operaciones como el 70% del total, dicho total asciende a 634400 operaciones a la semana y **la estructura estará compuesta por 26 grupos con dos miembros en cada grupo** para poder almacenar dicha ventana de redo log files con redundancia y que sobre espacio. Estos 26 grupos se distribuirán en 3 discos diferentes, almacenando en cada disco 18 miembros de los 52 que componen nuestra estructura de redo log files. Visualmente, la estructura queda definida de la siguiente forma:

DISCO 1	DISCO 2	DISCO 3
Miembro 1 de grupos [1-18]	Miembro 1 de grupos [19-26]	Miembro 2 de grupos [12-26]
	Miembro 2 de grupos [1-11]	

El mínimo de grupos necesarios es de 2 para la instalación del sistema en nuestro caso. Para conseguir los 26 grupos que necesitamos, deberemos comprobar cuántos grupos existen ya en el sistema y añadir los necesarios para lograr tener 26 mediante las siguientes sentencias oracle incluidas en un script:

Comprobamos cuántos grupos y miembros en ellos tenemos:

```
SELECT group#, member FROM v$logfile
```

Añadir un grupo:

```
ALTER database ADD LOGFILE (ruta-logFile/redo_numeroGrupo_numeroMiembro.log')  
size 1GB;
```

Por ejemplo:

```
ALTER database ADD LOGFILE (DISK01/redo_01_numero01.log') size 1GB;
```

Añadir un miembro a un grupo:

```
ALTER DATABASE ADD LOGFILE MEMBER  
'ruta-logfile/redo_numeroGrupo_numeroMiembro.log' TO GROUP numeroGrupo;
```

Por ejemplo:

```
ALTER DATABASE ADD LOGFILE MEMBER DISK01/redo_01_02.log' TO GROUP 1;
```

6. Estructura de Roles

Distinguiremos 4 roles. Un rol ejecutivo/gerente, el cual tendrá acceso de solo lectura a todos los tablespaces. Un rol trabajador, este tendrá acceso de lectura a las tablas correspondientes al tablespace de explotación. El tercero corresponderá a los usuarios partícipes en el desarrollo. Los cuales tendrán acceso total a los 3 tablespaces, desarrollo, explotación y pruebas. Por último el rol mantenimiento, que corresponderá a los informáticos responsables del sistema. Este último tendrá los mismos privilegios del rol anterior con el añadido que también tendrá permisos sobre el tablespace del system.

Ya que para dar permisos sobre las distintas tablas de los tablespaces necesitamos hacerlo de una en una, sería interesante tener un script que de al ejecutarlo diera permisos sobre todas las tablas del tablespace en concreto, ya que estos roles corresponden a la función que llevan a cabo estos usuarios sobre uno o varios tablespaces completos.

A continuación se muestra la creación de los distintos roles, y un ejemplo de los permisos dados para una tabla, aunque de nuevo, esto se debería hacer para todas las tablas que hubieran en ese tablespace.

Roles:

Trabajador:

```
CREATE ROLE trabajador IDENTIFIED EXTERNALLY;
```

```
GRANT SELECT  
ON ___EXPLOTACION.* ___  
TO ejecutivo;
```

Ejecutivo:

```
CREATE ROLE ejecutivo IDENTIFIED BY passwd1;
```

```
GRANT SELECT  
ON ___all.* ___  
TO ejecutivo;
```

Desarrolladores:

```
CREATE ROLE developer IDENTIFIED BY passwd2;
```

```
GRANT SELECT,  
INSERT,  
UPDATE,  
DELETE  
ON ___all.* ___  
TO developer;
```


Mantenimiento informático:

```
CREATE ROLE maintenance IDENTIFIED BY passwd3;
```

```
GRANT ALL PRIVILEGES  
ON _all_  
TO  
maintenance  
WITH ADMIN OPTIONS;
```

Perfiles: Siguiendo las indicaciones de la práctica en cuanto a las limitaciones y de los distintos tipos de usuarios y los posibles perfiles derivados de ellos tendremos 3 perfiles distintos. Tendremos los siguientes:

Gestión y ejecutivo: Se limitan las conexiones al sistema a 2.

```
CREATE PROFILE gestion_ejecutivo LIMIT SESSIONS_PER_USER 2;
```

informáticos y desarrolladores: estos no tienen ningún tipo de restricción y se le otorgaran recursos ilimitados:

```
CREATE PROFILE informatico  
SESSIONS_PER_USER          UNLIMITED  
CPU_PER_SESSION            UNLIMITED  
CPU_PER_CALL               UNLIMITED  
CONNECT_TIME               UNLIMITED  
IDLE_TIME                  UNLIMITED  
LOGICAL_READS_PER_SESSION  UNLIMITED  
LOGICAL_READS_PER_CALL     UNLIMITED  
PRIVATE_SGA                UNLIMITED  
COMPOSITE_LIMIT            UNLIMITED;
```

Usuarios de aplicaciones: Se limita el tiempo de conexión a 8 horas, que son 480 minutos

```
CREATE PROFILE usuario_aplicaciones CONNECT_TIME 480 LIMIT  
SESSIONS_PER_USER 1;
```

7. Copia de Seguridad

La entidad sólo puede realizar una parada total del sistema cada seis meses. Cada seis meses se hará una copia de seguridad en frío (backup físico), se copian los ficheros tal cual están en disco.

Las copias de seguridad irán aparte del sistema de manera que lo conectamos vía internet con fibra óptica y se abstraen del sistema y de las controladoras de nuestro uso.

Para la parada programada guardaremos de manera esencial:

- Datafiles (V\$DATAFILE: name, status) pero mejor (DBA_DATA_FILES: tablespace_name, file_name)
- Control files (V\$CONTROLFILE: name)
- Parameter files
- Password files
- Redo Log files (V\$LOGFILE: group#, member)

Copias en frío

En la cinta magnética 4 , dedicada solo para las copias con el sistema apagado

`mkdir /backup/frio`

`SQLPlus> SHUTDOWN IMMEDIATE;`

`SQLPlus> HOST cp -rvp /dev/sda*Para cada disco del 1 al 3 /dev/sdd/backup/frio`

** Con host indicamos que el sistema operativo anfitrión ejecute la orden cp para copiar, con los parámetros de -r (recursivo) , -v (verbose para informar de lo que hace) y -p (para guardar todas las propiedades).*

La copia la hacemos de las siguientes direcciones que guardan la totalidad del sistema

[disk 1 - 7]/databases/app/oracle y /databases/app/orainventory [disk 1 - 7]
SQLPlus> STARTUP;

Copias en caliente

Las copias en caliente se realizarán cada semana siguiendo al redo log file y se mantendrán 3 copias de seguridad en caliente. Necesitamos el Archivelog Mode parameter desde el fichero parameter la primera vez para que sea permanente.

Problemas con los Datafiles y Control files al estar en uso con las copias calientes

Copias de seguridad de control files

Backup en caliente de los tablespaces (si nadie está escribiendo)

Lo primero sería indicar al sistema el backup para poner el tablespace en modo solo lectura (Con el modo archivado activado): DISCOS DEL 4 - 7

- **ALTER TABLESPACE users BEGIN BACKUP;**

A continuación, usamos la herramientas del sistema operativo :

- **HOST cp -rvp /databases/app/oracle/oradata/oradba/users*.dbf /dev/sd[numero de cinta magnética correspondiente] /backup/caliente.**

Y para terminar terminamos el back up quitando el modo archivado:

- **ALTER TABLESPACE users END BACKUP;**

Y además una copia del fichero de control obligatoriamente aparte de las frecuencias de las copias de seguridad anteriores:

Cuando se realicen operaciones tales como:

- **ALTER DATABASE (ADD|DROP|RENAME|ARCHIVE...).**
- **ALTER TABLESPACE (ADD|DROP|DATAFILE...).**
- **DROP TABLESPACE.**

Comando : alter database backup controlfile to '/backups/encaliente/control1.bkp';

8. Referencias Bibliográficas

<https://docs.oracle.com/en/database/oracle/oracle-database>

Y mayoritariamente los seminarios y clases online impartidos a lo largo del curso