



UNIVERSIDAD DE GRANADA

ADMINISTRACIÓN DE BASES DE DATOS

Trabajo colaborativo

Leire Requena Garcia, Clara Maria Romero Lara

6 de junio de 2022

Índice general

1. Introducción	3
2. Estructura de almacenamiento	4
3. Creación de la base de datos	6
3.1. Base de datos	6
3.2. Tablespaces	7
3.3. Tablas	7
3.3.1. Inventario	7
3.3.2. Usuarios	8
3.3.3. Legal	8
3.3.4. Pruebas y desarrollo	9
4. Diseño de la estructura de perfiles y roles	10
4.1. Perfiles	10
4.1.1. Ejecutivo	10
4.1.2. Dev	10
4.1.3. Trabajo	10
4.2. Roles	11
4.2.1. Empleado	11
4.2.2. Desarrollo	11
4.2.3. Administración	11
5. Planificación de la copia de seguridad	12
5.1. Copia de seguridad de los datos	12
5.2. Replicación de los <i>control files</i>	12

Índice de figuras

1.1. Gráfico de tarta de los empleados de la empresa	3
2.1. Esquema de la estructura de discos de la base de datos	5
3.1. Diagrama entidad relacion de la base de datos	7

Parte 1:

Introducción

Durante la ejecución de esta práctica vamos a diseñar y desarrollar la estructura de la base de datos necesaria para la empresa *Smoliv S.L.*, una cooperativa aceitera que ha decidido vender sus productos artesanales en una tienda online con envíos a toda la península Ibérica.

Esta empresa consta 100 miembros y tiene la siguiente estructura:

- **Nivel ejecutivo:** 5 % de la empresa, en total 5 personas. Socios fundadores de la cooperativa dedicados a tareas de mando y administración general.
- **Nivel gerencial:** 20 % de la empresa, en total 20 personas. Empleados dedicados a labores de gestión de recursos humanos, contabilidad, cuestiones legales...
- **Nivel operativo:** 75 % de la empresa, en total 75 personas divididas en dos grupos:
 - **Técnicos informáticos:** 5 % de la empresa, en total 5 empleados. Encargados de mantener y desarrollar las aplicaciones y administrar la base de datos, dentro de este grupo al menos una persona tendrá privilegios de administración sobre todo el sistema.
 - **Empleados de otros departamentos:** 70 % de la empresa, en total 70 empleados. Trabajadores de diversas áreas que responden ante el nivel gerencial y usan las aplicaciones internas desarrolladas por los informáticos.

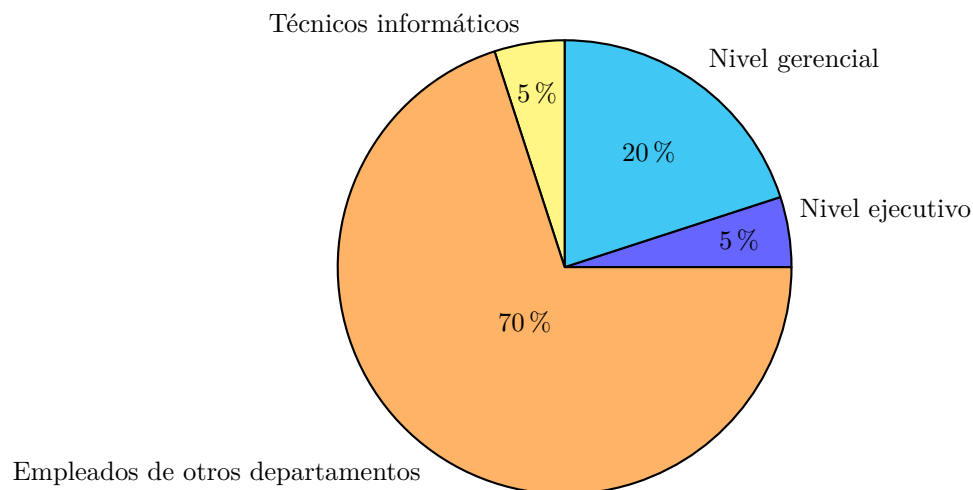


Figura 1.1: Gráfico de tarta de los empleados de la empresa

Parte 2:

Estructura de almacenamiento

Para simular la existencia de varios discos de almacenamiento físico distintos, al ser un supuesto teórico se estructurará la base de datos en directorios; cada uno contendrá *datafiles* relacionados con una parte de la empresa.

El modo de almacenamiento dependerá de las características de gestión, integridad y seguridad de cada uno de los datos; dividimos así en 5 categorías, 3 de explotación y dos de pruebas y desarrollo:

- **Inventario:** describe las existencias, salidas y entradas del género disponible en el almacén, además del material de oficina disponible para los empleados.
- **Usuarios:** información referida a todo usuario existente del sistema, ya sea un empleado de la empresa o un cliente de la tienda online.
- **Información legal y económica:** referida a todos los datos de cuestiones legales y económicas: nóminas, contratos, subvenciones...
- **Pruebas:** espacio de la base de datos dedicada a test del funcionamiento del sistema.
- **Desarrollo:** espacio dedicado a crear y actualizar las aplicaciones disponibles de la empresa.

Cada disco contendría un tablespace de un tamaño inicial de 500MB y sin límite de crecimiento.

Para el esquema de *Redo Log Files* vamos a usar 2 grupos [2], esto se debe a que es el número mínimo necesario y que no vamos a tener una alta afluencia de operaciones que puedan retrasar el sistema. Para su tamaño escogemos 512MB porque es una medida estandarizada de almacenamiento y en el caso de estos archivos interesa que el espacio desperdiciado sea mínimo [3]. Este espacio es más que suficiente porque aunque al ser una tienda online el flujo de consultas es teóricamente continuo, es un producto muy específico con un precio elevado que se mueve por picos y al estar localizada en España podemos reducir las horas más activas.

Basandonos en el número de operaciones por grupo semanales indicado en el supuesto, y suponiendo que un Redo Log File de 200MB almacena 5000 operaciones, podemos calcular el tamaño necesario para nuestro volumen de información:

$$\frac{(6 * 10^3 + 4 * 10^3 + 5 * 10^2 + 20 * 10^3) * 200MB}{5000} = 1220MB \quad (2.1)$$

Por lo tanto al usar 2 redos en cada disco, no todo ese volumen de información pertenece al mismo tipo de información. Si se diera el caso de que aumentase el volumen de operaciones podríamos analizar dónde se necesita y aumentar solo en ese lugar los archivos redo.

En conclusión la representación de discos quedaría de la siguiente forma:

- `/database/app/oracle/smoliv/inventario` Para el Inventario y sus redos
- `/database/app/oracle/smoliv/usuarios` Para usuarios y sus redos
- `/database/app/oracle/smoliv/legal` Para el apartado legal/económico y sus redos
- `/database/app/oracle/smoliv/pruebas` Para el espacio de pruebas del sistema
- `/database/app/oracle/smoliv/desarrollo` Para el espacio de desarrollo de aplicaciones
- `/database/app/oracle/smoliv/temp` Para ficheros temporales y de undo.

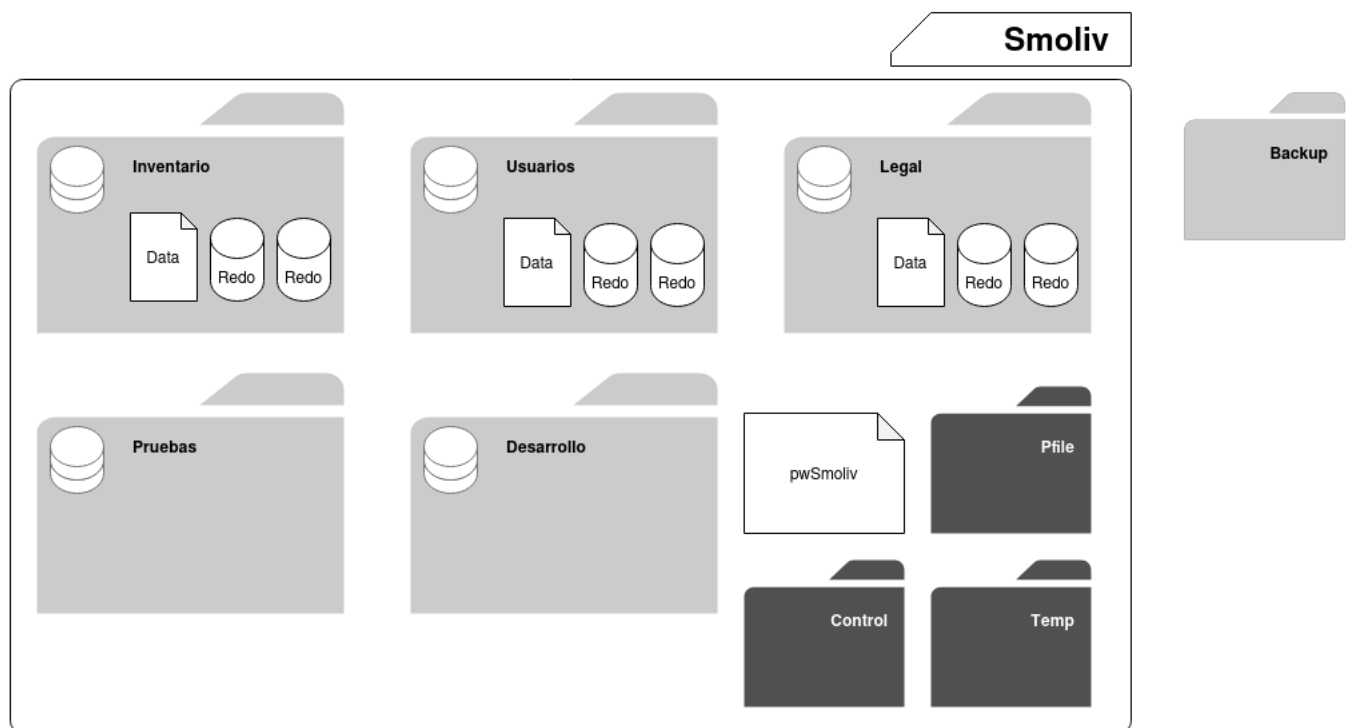


Figura 2.1: Esquema de la estructura de discos de la base de datos

Parte 3:

Creación de la base de datos

3.1: Base de datos

Para empezar hemos copiado el pfile que existía de la base de datos anterior y modificamos los nombres y rutas de acuerdo a las localizaciones que usaremos; esto es cambiar la ruta antigua de `$ORACLE_BASE` a nuestra carpeta `/database/app/oracle/smoliv`. Además creamos un archivo de contraseñas nuevo en el que creamos el usuario `sys` con la contraseña `aceit1_100`.

Con esto podemos iniciar la instancia con el siguiente comando, es importante haber iniciado sesión como `SYS`:

```
1 STARTUP NOMOUT PFILE=/database/app/oracle/smoliv/pfile/init.ora
```

En este estado es cuando podremos crear la base de datos y los tablespaces que necesitemos. La siguiente instrucción sirve para la primera tarea, crear la base de datos, hemos decidido usar 6 log files, dos por cada “disco” y usamos la orden que se encuentra en el seminario 5 de la asignatura.

```
1 CREATE DATABASE smoliv
2 USER sys IDENTIFIED BY "aceit1_100"
3 USER system IDENTIFIED BY "aceit1_100"
4 LOGFILE
5 GROUP 1 ('/database/app/oracle/smoliv/inventario/inventario_log1.log',
6         '/database/app/oracle/smoliv/inventario/inventario_log2.log') SIZE 200M,
7 GROUP 2 ('/database/app/oracle/smoliv/usuarios/usuarios_log1.log',
8         '/database/app/oracle/smoliv/usuarios/usuarios_log2.log') SIZE 200M,
9 GROUP 3 ('/database/app/oracle/smoliv/legal/legal_log1.log',
10        '/database/app/oracle/smoliv/legal/legal_log2.log') SIZE 200M
11 MAXLOGFILES 5
12 MAXLOGMEMBERS 5
13 MAXLOGHISTORY 1
14 MAXDATAFILES 100
15 MAXINSTANCES 1
16 CHARACTER SET us7ascii
17 NATIONAL CHARACTER SET al16utf16
18 EXTENT MANAGEMENT LOCAL
19 ENABLE PLUGGABLE DATABASE SEED
20 FILE_NAME_CONVERT=('/database/app/oracle/smoliv','/database/app/oracle/smoliv/')
21 SYSTEM DATAFILES SIZE 350M SYSAUX DATAFILES SIZE 100M
22 UNDO TABLESPACE undotbs1 DATAFILE '/database/app/oracle/smoliv/temp/undo.dbf' SIZE
23 200m
24 REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
25 USER_DATA TABLESPACE users DATAFILE '/database/app/oracle/smoliv/data/users.dbf' SIZE
26 500M;
```

3.2: Tablespaces

Como hemos mencionado, para simular varios discos usaremos directorios, por comodidad están dentro de la misma carpeta aunque hubiera sido más ideal sacarlos del directorio base. Tendremos un tablespace por cada grupo que definimos en la sección anterior 2.

```

1 CREATE TABLESPACE inventario DATAFILE '/database/app/oracle/smoliv/inventario/
  datainventario.dbf' SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
2
3 CREATE TABLESPACE usuarios DATAFILE '/database/app/oracle/smoliv/usuarios/datausuarios.
  dbf' SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
4
5 CREATE TABLESPACE legal DATAFILE '/database/app/oracle/smoliv/legal/datalegal.dbf' SIZE
  500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
6
7 CREATE TABLESPACE pruebas DATAFILE '/database/app/oracle/smoliv/pruebas/pruebas.dbf' SIZE
  500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
8
9 CREATE TABLESPACE desarrollo DATAFILE '/database/app/oracle/smoliv/desarrollo/desarrollo.
  dbf' SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

```

Hemos decidido el tamaño de datafile de 500M porque es una base de datos basada sobre todo en texto, además es una empresa pequeña y consideramos que es suficiente para el volumen de trabajo que creemos que tendría.

3.3: Tablas

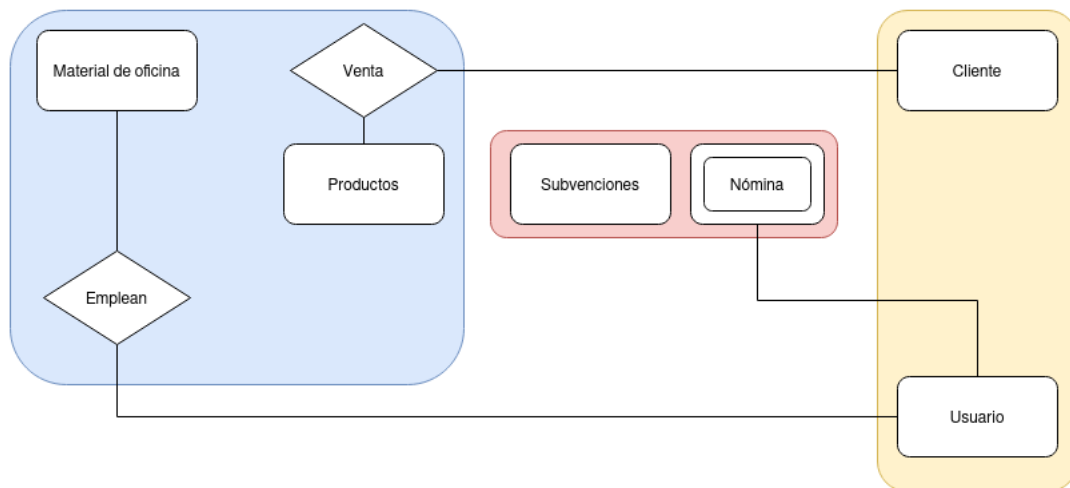


Figura 3.1: Diagrama entidad relacion de la base de datos

3.3.1: Inventario

El inventario guarda información de los productos en venta, las ventas realizadas, el material de oficina y quien emplea cada elemento; tendremos así cuatro tablas asociadas a este *tablespace*.

```

1 CREATE TABLE producto(
2     ID INT PRIMARY KEY NOT NULL,
3     nombre VARCHAR2(10),
4     cantidad INT,
5     precio INT NOT NULL
6 ) TABLESPACE inventario;

```



```

7
8 CREATE TABLE material_oficina(
9     nombre VARCHAR2(10) PRIMARY KEY NOT NULL,
10    departamento VARCHAR2(10),
11    cantidad INT
12 ) TABLESPACE inventario;
13
14 CREATE TABLE venta(
15     VENTA_ID INT PRIMARY KEY NOT NULL,
16     id_producto INT,
17     tienda VARCHAR2(10),
18     telefono CHAR(9),
19     FOREIGN KEY(id_producto) REFERENCES producto(ID),
20     FOREIGN KEY(telefono) REFERENCES cliente(TELEFONO)
21 ) TABLESPACE inventario;
22
23 CREATE TABLE emplea(
24     MATERIAL VARCHAR2(10) PRIMARY KEY NOT NULL,
25     ID_USUARIO CHAR(10) NOT NULL,
26     FOREIGN KEY(MATERIAL) REFERENCES material_oficina(nombre),
27     FOREIGN KEY(ID_USUARIO) REFERENCES usuario(DNI)
28 ) TABLESPACE inventario;

```

3.3.2: Usuarios

En este caso, tendremos dos tablas, una referida a los empleados de la empresa y otra para los clientes registrados en nuestra tienda online:

```

1 CREATE TABLE cliente(
2     TELEFONO CHAR(9) PRIMARY KEY NOT NULL,
3     nombre VARCHAR2(10),
4     apellidos VARCHAR2(20),
5     correo VARCHAR2(20),
6     direccion VARCHAR2(100)
7 ) TABLESPACE usuarios;
8
9 CREATE TABLE usuario(
10    DNI CHAR(10) PRIMARY KEY NOT NULL,
11    nombre VARCHAR2(10),
12    apellidos VARCHAR2(20),
13    correo VARCHAR2(20),
14    departamento VARCHAR2(10),
15    cuenta VARCHAR2(22)
16 ) TABLESPACE usuarios;

```

3.3.3: Legal

Guarda las nóminas de los trabajadores y las subvenciones recibidas, otra documentación legal no es relevante aquí.

```

1 CREATE TABLE nominas(
2     NUMERO_NOMINA INT PRIMARY KEY NOT NULL,
3     DNI_EMPLEADO CHAR(10),
4     salario NUMBER,
5     FOREIGN KEY(DNI_EMPLEADO) REFERENCES usuario(DNI)
6 ) TABLESPACE legal;
7
8 CREATE TABLE subvenciones(
9     NOMBRE VARCHAR(30) PRIMARY KEY NOT NULL,
10    fecha DATE, cantidad NUMBER
11 ) TABLESPACE legal;

```

3.3.4: Pruebas y desarrollo

En este caso, los *tablespaces* están vacíos ya que están pensados para ir aumentando según se programen las aplicaciones y se necesiten pruebas de errores lógicos y concurrencia, dejamos a los informáticos permisos de escritura y lectura para poder llevar a cabo su labor.

Parte 4:

Diseño de la estructura de perfiles y roles

4.1: Perfiles

Para cumplir con las indicaciones del supuesto tendremos tres perfiles, uno para cada nivel indicado anteriormente 1.

4.1.1: Ejecutivo

Otorgado a todos los usuarios del nivel ejecutivo y administrativo, tiene un límite de sesiones por usuario de 2 y ningún tiempo máximo establecido para terminar con la conexión:

```
1 CREATE PROFILE ejecutivo
2 LIMIT
3     SESSIONS_PER_USER 2
4     CONNECT_TIME UNLIMITED;
```

4.1.2: Dev

Se otorga a todos los usuarios del nivel operativo que sean técnicos informáticos, no tiene restricciones de número de conexiones simultáneas ni tiempo de conexión

```
1 CREATE PROFILE dev
2 LIMIT
3     SESSIONS_PER_USER UNLIMITED
4     CONNECT_TIME UNLIMITED;
```

4.1.3: Trabajador

Se otorga al resto de empleados de la empresa. Se organizan en dos turnos de 8 horas cada uno, por lo cual ese es el tiempo máximo de conexión. Destacamos que el parámetro de la consulta se mide en minutos, por lo cual, 8 horas corresponden a 480 minutos. Además limitamos las conexiones a una por persona.

```
1 CREATE PROFILE trabajador
2 LIMIT
3     SESSIONS_PER_USER 1
4     CONNECT_TIME 480;
```

4.2: Roles

Hemos considerado que para gestionar la base de datos necesitamos 3 roles, que no son indicativos por si solos del nivel de responsabilidad que asumen en la jerarquía de la empresa.

4.2.1: Empleado

Se concede a todos los empleados de la empresa, este rol permite acceso a los datos dedicados a explotación de la empresa, los que se almacenan en los *tablespaces* inventario, usuarios y legal. Los privilegios que posee el rol son sobre los objetos de estos espacios, con la excepción de borrar datos y/o tablas que consideramos que debería llevarse a cabo por un técnico informático para no interrumpir las actividades de estos últimos.

```
1 CREATE ROLE empleado;
2
3 GRANT SELECT, INSERT, UPDATE ON producto TO empleado;
4 GRANT SELECT, INSERT, UPDATE ON material_oficina TO empleado;
5 GRANT SELECT, INSERT, UPDATE ON venta TO empleado;
6 GRANT SELECT, INSERT, UPDATE ON cliente TO empleado;
7 GRANT SELECT, INSERT, UPDATE ON usuario TO empleado;
8 GRANT SELECT, INSERT, UPDATE ON nominas TO empleado;
9 GRANT SELECT, INSERT, UPDATE ON subvenciones TO empleado;
```

4.2.2: Desarrollo

Este rol está designado para los técnicos informáticos que emplean los *tablespaces* de prueba y desarrollo; además de estos también tienen acceso al resto de la base de datos por la cláusula **ANY**. Necesitan por tanto permisos que les permitan crear y modificar los objetos existentes dentro del sistema.

```
1 CREATE ROLE desarrollo;
2
3 GRANT CREATE ANY TABLE, ALTER ANY TABLE, DROP ANY TABLE,
4     CREATE ANY VIEW, DROP ANY VIEW,
5     CREATE ANY TRIGGER, ALTER ANY TRIGGER, DROP ANY TRIGGER,
6     CREATE ANY SEQUENCE, DROP ANY SEQUENCE,
7     CREATE ANY SYNONYM, DROP ANY SYNONYM,
8     CREATE ANY PROCEDURE, DROP ANY PROCEDURE, DEBUG ANY PROCEDURE
9 TO desarrollo;
```

4.2.3: Administración

Este rol no está pensado ni para el nivel ejecutivo ni el administrativo, se concede a unas pocas personas (cuantas menos mejor) del departamento informático ya que otorga privilegios ilimitados sobre toda la estructura de la base de datos, incluyendo consultas y operaciones potencialmente destructivas.

```
1 CREATE ROLE administracion;
2
3 GRANT DBA TO administracion;
```

Parte 5:

Planificación de la copia de seguridad

5.1: Copia de seguridad de los datos

Dado que en el supuesto se indica que solo es posible hacer una parada total del sistema cada 6 meses, se realizará en ese momento una copia de seguridad de toda la base de datos. Elegimos el método on-line que reduce el tiempo que la base de datos está completamente caída, además sería buena idea realizar esta tarea en un momento de bajo flujo de trabajo, como en altas horas de la madrugada.

Para realizar la copia de seguridad ejecutamos la siguiente lista de operaciones:

```
1 SHUTDOWN IMMEDIATE;
2 STARTUP MOUNT;
3 ALTER DATABASE ARCHIVELOG;
4 ALTER DATABASE OPEN;
5
6 ALTER TABLESPACE inventario BEGIN BACKUP;
7 ALTER TABLESPACE usuarios BEGIN BACKUP;
8 ALTER TABLESPACE legal BEGIN BACKUP;
9 ALTER TABLESPACE pruebas BEGIN BACKUP;
10 ALTER TABLESPACE desarrollo BEGIN BACKUP;
11
12 HOST CP -rvp /database/app/oracle/smoliv/inventario/datainventario*.dbf
13 /database/app/backup
14 HOST CP -rvp /database/app/oracle/smoliv/usuarios/datausuarios*.dbf
15 /database/app/backup
16 HOST CP -rvp /database/app/oracle/smoliv/legal/datalegal*.dbf
17 /database/app/backup
18 HOST CP -rvp /database/app/oracle/smoliv/pruebas/pruebas*.dbf
19 /database/app/backup
20 HOST CP -rvp /database/app/oracle/smoliv/desarrollo/desarrollo*.dbf
21 /database/app/backup
22
23 ALTER TABLESPACE inventario END BACKUP;
24 ALTER TABLESPACE usuarios END BACKUP;
25 ALTER TABLESPACE legal END BACKUP;
26 ALTER TABLESPACE pruebas END BACKUP;
27 ALTER TABLESPACE desarrollo END BACKUP;
```

Podemos automatizar este proceso creando un script y activándolo en el archivo `/etc/crontab` y que sea el programa `cron` quien se encargue de ejecutar las sentencias.

5.2: Replicación de los *control files*

Los *control file* son archivos binarios que almacenan la estructura física de nuestra base de datos, incluyendo su nombre, localización de *datafiles* y *redo logs*, fecha de creación, el número de log actual e información del *checkpoint* [1].

El *control file* debe estar disponible cada vez que iniciemos la base de datos, si no, no podremos montarla y su recuperación es difícil. Por defecto Oracle crea una copia de este archivo, aún así es recomendable hacer una

copia manual en un disco separado cada cierto tiempo. Como para esto hay que detener el sistema, podemos realizar la replicación cada 6 meses junto con la copia de seguridad de los datos.

En concreto, usamos la siguiente sentencia:

```
1 ALTER DATABASE BACKUP controlfile TO '/databases/app/oracle/backup/control.bkp';
```

Bibliografía

- [1] Oracle. Backing up control files. https://docs.oracle.com/cd/B19306_01/server.102/b14231/control.htm#i1006485.
- [2] Oracle. Managing the redo log. https://docs.oracle.com/cd/B19306_01/server.102/b14231/onlineredo.htm.
- [3] Oracle. Planning the size of redo log files. https://docs.oracle.com/cd/E18283_01/server.112/e17120/onlineredo002.htm.