



12.5 Gestión de seguridad

La gestión de seguridad tiene mucho que ver con la gestión de usuarios y con la concepción y supresión de privilegios a los usuarios. El administrador de la base de datos es el responsable de permitir o denegar el acceso a los usuarios a determinados objetos o recursos de la base de datos.

Podemos clasificar la seguridad de la base de datos en dos categorías: *seguridad del sistema* y *seguridad de los datos*.

- La **seguridad del sistema** incluye los mecanismos que controlan el acceso y uso de la base de datos a nivel del sistema. Por ejemplo: cada vez que se conecta un usuario a la base de datos, los mecanismos de seguridad comprobarán si éste está autorizado.
- La **seguridad de los datos** incluye mecanismos que controlan el acceso y uso de la base de datos a nivel de objetos. Por ejemplo, cada vez que un usuario acceda a un objeto (una tabla, una vista, etcétera), los mecanismos de seguridad comprobarán si dicho usuario puede acceder a ese objeto y qué tipo de operación puede hacer con él (SELECT, INSERT, etcétera).

A. Usuarios

Un **usuario** es un nombre definido en la base de datos que se puede conectar a ella y acceder a determinados objetos según ciertas condiciones que define el administrador. Para acceder a la base de datos, los usuarios deben ejecutar una aplicación de base de datos, como SQL*Plus, Oracle Forms y conectarse usando el nombre definido en la base de datos. También pueden acceder a través de un navegador web.

Asociado con cada usuario de la base de datos hay un esquema con el mismo nombre. Un **esquema** es una colección lógica de objetos (tablas, vistas, secuencias, sinónimos, índices, clusters, procedures, funciones, paquetes, etcétera). Por defecto, cada usuario tiene acceso a todos los objetos de su esquema correspondiente, y puede acceder a los objetos de otro usuario siempre y cuando este otro le haya concedido el privilegio de hacerlo. Veamos cómo se crean, se modifican y borran los usuarios.

- **Creación de usuarios**

Al instalar la base de datos Oracle se crean automáticamente dos usuarios con el privilegio de administrador de la base de datos (DBA). Estos usuarios son SYS y SYSTEM. Durante el proceso de instalación se pedirá la clave para cada usuario.

El usuario SYS es el propietario de las tablas del diccionario de datos; en ellas se almacena información sobre el resto de las estructuras de la base de datos. Oracle maneja las tablas del usuario SYS; ningún usuario, aunque sea administrador, puede modificar las tablas de SYS. Sólo nos conectaremos como usuario SYS cuando las instrucciones de Oracle lo exijan.

El diccionario de datos está formado por un conjunto de tablas y vistas en el *tablespace* SYSTEM. Los usuarios tienen acceso de sólo lectura a las vistas de este diccionario, el cual se crea a la vez que la base de datos y es propiedad del usuario SYS. Contiene: objetos de la base

Estos mecanismos de seguridad no son más que privilegios del sistema, privilegios sobre objetos, roles, límites a los recursos, asignación de espacio para almacenar objetos, etc.



12. Administración de Oracle I

12.5 Gestión de seguridad

de datos, nombres de usuario, derechos y autorizaciones, restricciones, información sobre el espacio libre/ocupado, información de exportación e información sobre otros objetos.

Los objetos del diccionario de datos a los que un usuario puede acceder se encuentran en la vista DICTIONARY, que es propiedad del usuario SYS. Con la orden SQL> SELECT TABLE_NAME FROM DICTIONARY; se visualizan los objetos del diccionario de datos a los que se puede acceder. El prefijo de una vista del diccionario indica el nivel de acceso a él:

Vistas USER y ALL

Accesibles para todos los usuarios.

Vistas DBA

Sólo el administrador puede utilizar estas vistas.

El usuario SYSTEM es creado por Oracle para realizar las tareas de administración de la base de datos. No se suelen crear tablas de usuario en el esquema de SYSTEM. Para crear otros usuarios es preciso conectarse como usuario SYSTEM, ya que éste posee el correspondiente privilegio. Al instalar Oracle, el administrador de la base de datos ha de crearse un usuario para sí mismo con los derechos de administrador y realizar todas las tareas de administración con este nombre de usuario.

Para crear usuarios se necesita el privilegio CREATE USER. La orden para crear usuarios es CREATE USER, cuyo formato es:

```
CREATE USER nombre_usuario  
IDENTIFIED BY clave_acceso  
[DEFAULT TABLESPACE espacio_tabla]  
[TEMPORARY TABLESPACE espacio_tabla]  
[QUOTA {entero {K|M} |UNLIMITED} ON espacio_tabla]  
[PROFILE perfil];
```

Donde:

- La primera opción, CREATE USER, crea un nombre de usuario que será identificado por el sistema.
- La segunda opción, IDENTIFIED BY, permite dar una clave de acceso al usuario creado. Por ejemplo, para crear el usuario MILAGROS, con clave de acceso MILAGROS, utilizaremos la siguiente orden: CREATE USER MILAGROS IDENTIFIED BY MILAGROS; Pero si MILAGROS intenta conectarse a Oracle no podrá porque no tiene privilegios para iniciar sesión en la base de datos. Oracle devuelve un mensaje de error al intentar conectar:

```
SQL> CONNECT  
Introduzca el nombre de usuario: MILAGROS  
Introduzca la contraseña: *****  
ERROR:  
ORA-01045: user MILAGROS lacks CREATE SESSION privilege;  
logon denied
```

- DEFAULT TABLESPACE asigna a un usuario el *tablespace* por defecto para almacenar los objetos que cree. Si no se asigna ninguno, el *tablespace* por defecto será USERS.
- TEMPORARY TABLESPACE especifica el nombre de *tablespace* para trabajos temporales. Si no se especifica ninguno, el *tablespace* por defecto es TEMP.



- QUOTA asigna un espacio en megabytes o en kilobytes en el *tablespace* asignado. Si no se especifica esta cláusula, el usuario no tiene cuota asignada y no podrá crear objetos en el *tablespace*. Si un usuario dispone de acceso y de recursos ilimitados a cualquier *tablespace*, se le debe dar el privilegio UNLIMITED TABLESPACE con la orden GRANT, que se verá después.
- PROFILE asigna un perfil al usuario. Si se omite, Oracle asigna el perfil por omisión al usuario. Un perfil limita el número de sesiones concurrentes de usuario, limita el tiempo de uso de la CPU, el tiempo de una sesión, desconecta al usuario si sobrepasa el tiempo, etcétera.

Caso práctico



- 4 La siguiente orden crea un usuario de nombre USUARIO1. La clave es la misma. El *tablespace* por omisión es USERS (ya que no se indica) al cual se han asignado 500 kilobytes. El *tablespace* para trabajos temporales es TEMP (ya que no se indica en la orden):

```
CREATE USER USUARIO1 IDENTIFIED BY USUARIO1 QUOTA 500K ON USERS;
```

La siguiente orden crea un usuario de nombre USUARIO2, la clave es la misma. El *tablespace* por omisión es TRABAJO al cual se han asignado 1 megabyte. El *tablespace* para trabajos temporales es TEMPORAL al cual se han asignado 500 kilobytes:

```
CREATE USER USUARIO2 IDENTIFIED BY USUARIO2 DEFAULT TABLESPACE TRABAJO TEMPORARY
TABLESPACE TEMPORAL QUOTA 1M ON TRABAJOS QUOTA 500K ON TEMPORAL;
```

Vistas con información de usuarios:

- **USER_USERS:** Obtiene información del usuario actual: la fecha de creación, los *tablespaces* asignados, el identificador, etcétera.
- **ALL_USERS:** Obtiene información acerca de todos los usuarios creados en la base de datos: el nombre, la fecha de creación y su identificador.

Modificación de usuarios

Las opciones dadas a un usuario en la orden CREATE USER se pueden modificar con la orden ALTER USER. Es posible cambiar la clave de acceso, el *tablespace* por defecto, el *tablespace* temporal, la cuota en los *tablespaces* o el perfil. El formato es el siguiente:

```
ALTER USER nombre_usuario
IDENTIFIED BY clave_acceso
[DEFAULT TABLESPACE espacio_tabla]
[TEMPORARY TABLESPACE espacio_tabla]
[QUOTA {entero {K|M} |UNLIMITED} ON espacio_tabla]
[PROFILE perfil];
```

Los parámetros significan lo mismo que en la orden CREATE.



12. Administración de Oracle I

12.5 Gestión de seguridad

Cada usuario puede cambiar únicamente su clave de acceso. No puede cambiar el *tablespace* por defecto, la cuota en los *tablespaces* ni el perfil, a no ser que tenga el privilegio ALTER USER. Por ejemplo, el USUARIO01 cambia su clave: **ALTER USER USUARIO01 IDENTIFIED BY NUEVACLAVE;**

Borrado de usuarios

Podemos borrar un usuario de la base de datos, incluidos los objetos que contiene. Para borrar usuarios se usa la orden DROP USER, que tiene este formato:

```
DROP USER usuario [CASCADE];
```

La opción CASCADE suprime todos los objetos del usuario antes de borrar el usuario.

Para poder borrar usuarios es preciso tener el privilegio DROP USER.

Por ejemplo, borramos el USUARIO01: **DROP USER USUARIO01.** Si el usuario tiene objetos creados se visualiza un mensaje de error: ORA-01922:

Se debe especificar CASCADE para borrar USUARIO01 y no nos deja borrarlo, tenemos que añadir la opción CASCADE: **DROP USER USUARIO01 CASCADE;**

B. Privilegios

Un **privilegio** es la capacidad de un usuario dentro de la base de datos a realizar determinadas operaciones o a acceder a determinados objetos de otros usuarios. Ningún usuario puede llevar a cabo una operación si antes no se le ha concedido permiso.

Mediante la asignación de privilegios se permite o restringe el acceso a los datos o la realización de cambios en los datos, la posibilidad de realizar funciones del sistema, etcétera.

Cuando se crea un usuario es necesario darle privilegios para que pueda hacer algo. Oracle ofrece varios roles o funciones, tres de ellos se aplican al entorno de desarrollo: CONNECT, RESOURCE y DBA. Los demás están relacionados con la administración de la base de datos: EXP_FULL_DATABASE, IMP_FULL_DATABASE, DELETE_CATALOG_ROLE, DM_CATALOG_ROLE, HS_ADMIN_ROLE, AQ_USER_ROLE, etcétera.

Un rol o función está formado por un conjunto de privilegios. A continuación, se exponen los privilegios para cada uno de los roles que se aplican al entorno de desarrollo:

Roles (funciones)	Privilegios
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE y CREATE VIEW.
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE TABLE, CREATE SEQUENCE, CREATE TRIGGER y CREATE TYPE
DBA	Posee todos los privilegios del sistema



Hay dos tipos de privilegios que podemos definir en la base de datos: privilegios sobre los objetos y privilegios del sistema.

- **Privilegios sobre los objetos**

Estos privilegios nos permiten acceder y realizar cambios en los datos de los objetos de otros usuarios.

Por ejemplo, el privilegio de consultar la tabla de otro usuario es un privilegio sobre objetos.

Se dispone de los siguientes privilegios sobre los objetos tablas, vistas, secuencias y *procedures*:

Privilegio sobre los objetos	Tabla	Vista	Secuencia	Procedure
ALTER	X			
DELETE	X			
EXECUTE		X		
INDEX	X			
INSERT	X			
REFERENCES	X			
SELECT	X		X	
UPDATE	X		X	

La orden para dar privilegios sobre los objetos es **GRANT**, con el siguiente formato:

```
GRANT {priv_objeto [,priv_objeto] ...|ALL [PRIVILEGES] }
      [(columna [,columna]...)]
  ON   [usuario.]objeto
  TO    {usuario|rol|PUBLIC [, {usuario|rol|PUBLIC} ...]
  [WITH GRANT OPTION];
```

Donde:

- **ON** especifica el objeto sobre el que se dan los privilegios.
- **TO** identifica a los usuarios o roles a los que se conceden los privilegios.
- **ALL** concede todos los privilegios sobre el objeto especificado.
- La cláusula **WITH GRANT OPTION** permite que el receptor del privilegio o rol se lo asigne a otros usuarios o roles.
- **PUBLIC** asigna los privilegios a todos los usuarios actuales y futuros. El propósito principal del grupo PUBLIC es garantizar el acceso a determinados objetos a todos los usuarios de la base de datos.

Con la orden GRANT se pueden conceder privilegios INSERT, UPDATE o REFERENCES sobre determinadas columnas de una tabla.



12. Administración de Oracle I

12.5 Gestión de seguridad



Caso práctico

- 5 El usuario MILAGROS tiene una tabla de nombre TABLA1 que contiene la temperatura de una serie de ciudades. Concede a FRANCISCO los privilegios de SELECT e INSERT en TABLA1: GRANT SELECT, INSERT ON TABLA1 TO FRANCISCO; Ahora FRANCISCO puede acceder a la TABLA1 de MILAGROS de la siguiente manera: SELECT * FROM MILAGROS.TABLA1;

El usuario MILAGROS da privilegios a JUAN para insertar en TABLA1: GRANT INSERT ON TABLA1 TO JUAN; Ahora Juan sólo puede insertar en la TABLA1 de MILAGROS, si intenta hacer SELECT se producirá un error:

```
SELECT * FROM MILAGROS.TABLA1;
*
ERROR en línea 1:
ORA-01031: privilegios insuficientes
```

MILAGROS concede a FRANCISCO todos los privilegios sobre TABLA1: GRANT ALL ON TABLA1 TO FRANCISCO;

MILAGROS concede todos los privilegios sobre TABLA1 a todos los usuarios, incluyendo a los que se crean después de ejecutar esta orden: GRANT ALL ON TABLA1 TO PUBLIC;

MILAGROS concede a JUAN sobre TABLA1 para que pueda modificar sólo la columna temperatura: GRANT UPDATE (TEMPERATURA) ON TABLA1 TO JUAN; Si JUAN intenta modificar las dos columnas de la tabla, evidentemente, no podrá. En cambio, no tendrá ningún problema al modificar la columna a la cual tiene acceso: UPDATE MILAGROS.TABLA1 SET TEMPERATURA=20 WHERE CIUDAD='MADRID' ;

Con la opción **WITH GRANT OPTION** se puede dar al usuario que recibe los privilegios el privilegio para que él pueda concederlos a otros. Ahora MILAGROS concede a FRANCISCO el privilegio para insertar en TABLA1 y, además, para que él pueda pasar este privilegio a otros usuarios: GRANT INSERT ON TABLA1 TO FRANCISCO WITH GRANT OPTION;

Ahora FRANCISCO puede conceder el privilegio INSERT a otros usuarios sobre la tabla TABLA1 de MILAGROS. A continuación, da el privilegio INSERT sobre TABLA1 a JUAN: GRANT INSERT ON MILAGROS.TABLA1 TO JUAN;



Actividades propuestas

- 1 Concede el privilegio SELECT e INSERT sobre la tabla DEPART a uno de tus compañeros de clase con la opción de que se lo pueda conceder a otros.

Concede el privilegio UPDATE sobre la columna APELLIDO de la tabla EMPLE a un compañero de clase.

Prueba los privilegios recibidos sobre las tablas.

12. Administración de Oracle I

12.5 Gestión de seguridad



• Privilegios del sistema

Son los que dan derecho a ejecutar un tipo de comando SQL o a realizar alguna acción sobre objetos de un tipo especificado. Por ejemplo, el privilegio para crear *tablespaces* es un privilegio del sistema. Existen más de 80 tipos de privilegios distintos disponibles. Algunos de ellos son los siguientes:

PRIVILEGIO DEL SISTEMA	OPERACIONES AUTORIZADAS
CREATE ANY INDEX	INDEX Crear un índice en cualquier esquema, en cualquier tabla.
ALTER ANY INDEX	Modificar cualquier índice de la base de datos.
DROP ANY INDEX	Borrar cualquier índice de la base de datos.
GRANT ANY PRIVILEGE	PRIVILEGE Conceder cualquier privilegio de sistema.
CREATE ANY PROCEDURE	PROCEDURE Crear procedimientos almacenados, funciones y paquetes en cualquier esquema.
CREATE PROCEDURE	Crear procedimientos almacenados, funciones y paquetes en nuestro esquema.
ALTER ANY PROCEDURE	Modificar procedimientos almacenados, funciones y paquetes en cualquier esquema.
DROP ANY PROCEDURE	Borrar procedimientos almacenados, funciones y paquetes en cualquier esquema.
EXECUTE ANY PROCEDURE	Ejecutar procedimientos, funciones o referencias a paquetes públicos en cualquier esquema.
CREATE PROFILE	PROFILE Crear un perfil de usuario.
ALTER PROFILE	Modificar cualquier perfil.
DROP PROFILE	Borrar cualquier perfil.
CREATE ROLE	ROLE Crear roles.
ALTER ANY ROLE	Modificar roles.
DROP ANY ROLE	Borrar cualquier rol.
GRANT ANY ROLE	Dar permisos para cualquier rol de la base.
CREATE SEQUENCE	SEQUENCE Crear secuencias en nuestro esquema.
ALTER ANY SEQUENCE	Modificar cualquier secuencia de la base.
DROP ANY SEQUENCE	Borrar secuencias de cualquier esquema.
SELECT ANY SEQUENCE	Referenciar secuencias de cualquier esquema.
CREATE SESSION	SESSION Conectarnos a la base de datos.
ALTER SESSION	Manejar la orden ALTER SESSION.
RESTRICTED SESSION	Conectarnos a la base de datos cuando se ha levantado con STARTUP RESTRICT.
CREATE SYNONYM	SYNONYM Crear sinónimos en nuestro esquema.
CREATE PUBLIC SYNONYM	Crear sinónimos públicos.
DROP PUBLIC SYNONYM	Borrar sinónimos públicos.
CREATE ANY SYNONYM	Crear sinónimos en cualquier esquema.
DROP ANY SYNONYM	Borrar sinónimos de cualquier esquema.
CREATE TABLE	TABLE Crear tablas en nuestro esquema y generar índices sobre las tablas del esquema.
CREATE ANY TABLE	Crear una tabla en cualquier esquema.
ALTER ANY TABLE	Modificar una tabla en cualquier esquema.

(Continúa)



12. Administración de Oracle I

12.5 Gestión de seguridad

(Continuación)

PRIVILEGIO DEL SISTEMA	OPERACIONES AUTORIZADAS
DROP ANY TABLE	Borrar una tabla en cualquier esquema.
LOCK ANY TABLE	Bloquear una tabla en cualquier esquema.
SELECT ANY TABLE	Hacer SELECT en cualquier tabla.
INSERT ANY TABLE	Insertar filas en cualquier tabla.
UPDATE ANY TABLE	Modificar filas en cualquier tabla.
DELETE ANY TABLE	Borrar filas de cualquier tabla.
	TABLESPACES
CREATE TABLESPACE	Crear espacios de tablas.
ALTER TABLESPACE	Modificar tablespaces.
MANAGE TABLESPACES	Poner on-line u off-line a cualquier tablespace.
DROP TABLESPACE	Eliminar tablespaces.
UNLIMITED TABLESPACE	Utilizar cualquier espacio de cualquier tablespace.
	TYPE
CREATE TYPE	Crea tipos de objeto y cuerpos de tipos de objeto en el propio esquema.
CREATE ANY TYPE	Crea tipos de objeto y cuerpos de tipos de objeto en cualquier esquema.
ALTER ANY TYPE	Modifica tipos de objeto en cualquier esquema.
DROP ANY TYPE	Elimina tipos de objeto y cuerpos de tipos de objeto en cualquier esquema.
EXECUTE ANY TYPE	Utiliza y hace referencia a tipos de objeto y tipos de colección en cualquier esquema.
UNDER ANY TYPE	Crea subtipos a partir de cualquier tipo de objeto no final.
	USER
CREATE USER	Crear usuarios y crear cuotas sobre cualquier espacio de tablas, establecer espacios de tablas por omisión y temporales.
ALTER USER	Modificar cualquier usuario. Este privilegio autoriza al que lo recibe a cambiar la contraseña de otro usuario, a cambiar cuotas sobre cualquier espacio de tablas, a establecer espacios de tablas por omisión, etcétera.
DROP USER	Eliminar usuarios.
	VIEW
CREATE VIEW	Crear vistas en el esquema propio.
CREATE ANY VIEW	Crear vistas en cualquier esquema.
DROP ANY VIEW	Borrar vistas en cualquier esquema.
	OTROS
SYSDBA	Ejecutar operaciones STARTUP y SHUTDOWN, ALTER DATABASE, CREATE DATABASE, ARCHIVELOG y RECOVERY, CREATE SPFILE
SYSOPER	Ejecutar operaciones STARTUP y SHUTDOWN, ALTER DATABASE, ARCHIVELOG y RECOVERY, CREATE SPFILE

El formato de la orden GRANT para asignar privilegios del sistema es:

```
GRANT {privilegio|rol} [, {privilegio|rol}, ...]
TO{usuario|rol|PUBLIC}[,{usuario|rol|PUBLIC}] ...
[WITH ADMIN OPTION];
```

Donde:

- TO identifica a los usuarios o roles a los que se conceden los privilegios.
- La cláusula WITH ADMIN OPTION permite que el receptor del privilegio o rol pueda conceder esos mismos privilegios a otros usuarios o roles.



Caso práctico

- 6 Cuando creamos un usuario tenemos que darle privilegios para que, como mínimo, pueda iniciar sesión en la base de datos. Creamos el usuario PEDRO y le damos el privilegio de crear sesión (CREATE SESSION): CREATE USER PEDRO IDENTIFIED BY PEDRO QUOTA 500K ON USERS;

```
GRANT CREATE SESSION TO PEDRO;
```

Se concede a PEDRO el rol CONNECT, lo que le permitirá tener todos los privilegios descritos para este rol (ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE y CREATE VIEW): GRANT CONNECT TO PEDRO;

Ahora se concede a PEDRO y a USUARIO1 el privilegio de administrador del sistema (DBA): GRANT DBA TO PEDRO, USUARIO1;

Para hacer que MILAGROS pueda borrar usuarios y, además, pueda conceder este privilegio a otros usuarios, se utiliza la opción WITH ADMIN OPTION: GRANT DROP USER TO MILAGROS WITH ADMIN OPTION;

Para hacer que todos los usuarios puedan hacer SELECT en cualquier tabla de cualquier usuario, escribimos: GRANT SELECT ANY TABLE TO PUBLIC;

- **Retirada de privilegios**

Al igual que se conceden privilegios, se pueden retirar. Para eso sirve la orden SQL REVOKE, que retira privilegios o roles concedidos a los usuarios y privilegios concedidos a los roles. El formato para retirar privilegios de objetos a los usuarios o roles es:

```
REVOKE {priv_objeto [,priv_objeto] ...|ALL [PRIVILEGES]}  
ON [usuario.]objeto  
FROM {usuario|rol|PUBLIC} [, {usuario|rol|PUBLIC}]...;
```

Y el formato para retirar privilegios de sistema o roles a usuarios o para retirar privilegios a roles es el siguiente:

```
REVOKE {priv_sistema|rol} [, {priv_sistema|rol}] ...  
FROM {usuario|rol|PUBLIC} [, {usuario|rol|PUBLIC}]...;
```

Caso práctico

- 7 MILAGROS retira los privilegios SELECT y UPDATE sobre TABLA1 a FRANCISCO: REVOKE SELECT, UPDATE ON TABLA1 FROM FRANCISCO;

(Continúa)



12. Administración de Oracle I

12.5 Gestión de seguridad

(Continuación)

REVOKE ALL elimina todos los privilegios concedidos anteriormente sobre algún objeto. La opción WITH GRANT OPTION desaparece con el privilegio con el cual fue asignada. En este ejemplo, MILAGROS retira todos los privilegios concedidos a FRANCISCO y JUAN sobre TABLA1: **REVOKE ALL ON TABLA1 FROM FRANCISCO, JUAN;**

Retiramos el privilegio de borrar usuarios a MILAGROS: **REVOKE DROP USER FROM MILAGROS;**

Retiramos el privilegio de consultar cualquier tabla a todos los usuarios: **REVOKE SELECT ANY TABLE FROM PUBLIC;**

Retiramos el privilegio de administrador (DBA) a los usuarios JUAN y PEDRO: **REVOKE DBA FROM JUAN, PEDRO;**

• Vistas con información de los privilegios

Para conocer los privilegios que han concedido o recibido los usuarios sobre los objetos o a nivel del sistema, podemos consultar las siguientes vistas del diccionario de datos:

- SESSION_PRIVS: privilegios del usuario activo.
- USER_SYS_PRIVS: privilegios de sistema asignados al usuario.
- DBA_SYS_PRIVS: privilegios de sistema asignados a los usuarios o a los roles.
- USER_TAB_PRIVS: concesiones sobre objetos que son propiedad del usuario, concedidos o recibidos por éste.
- USER_TAB_PRIVS_MADE: concesiones sobre objetos que son propiedad del usuario (asignadas).
- USER_TAB_PRIVS_RECV: concesiones sobre objetos que recibe el usuario.
- USER_COL_PRIVS: concesiones sobre columnas en las que el usuario es el propietario, asigna el privilegio o lo recibe.
- USER_COL_PRIVS_MADE: todas las concesiones sobre columnas de objetos que son propiedad del usuario.
- USER_COL_PRIVS_RECV: concesiones sobre columnas recibidas por el usuario.



Actividades propuestas

- 2 Escribe una secuencia de órdenes en la que se crea un usuario, se le asigna el privilegio de iniciar sesión en Oracle y de crear una tabla. Despues conéctate con ese usuario y consulta los privilegios del sistema que tiene. Concede permiso SELECT sobre la tabla creada a otro usuario y consulta los privilegios sobre objetos concedidos y recibidos por él.



C. Roles

Supongamos que un conjunto de usuarios del departamento de contabilidad requiere el mismo conjunto de privilegios para trabajar con ciertos datos. Este conjunto de privilegios se puede agrupar en un rol, de tal manera que es posible asignar el mismo rol a cada uno de los usuarios. Un **rol** o **función** es un conjunto de privilegios que recibe un nombre común para facilitar la tarea de asignación de éstos a los usuarios o a otros roles. Los privilegios de un rol pueden ser de sistema y a nivel de objeto. En primer lugar creamos el rol con la orden SQL CREATE ROLE y, a continuación, asignamos privilegios con la orden GRANT. El formato para crear un rol es:

```
CREATE ROLE NombreRol;
```

Para crear un rol se requiere el privilegio de sistema CREATE ROLE. Por ejemplo, creamos un rol llamado ACCESO: **CREATE ROLE ACCESO;**

Conceder privilegios a los roles

Una vez creado, hemos de concederle privilegios usando la orden GRANT. Por ejemplo, asignamos los privilegios al rol ACCESO: SELECT e INSERT sobre la tabla EMPLE, INSERT en la tabla DEPART, y CREATE SESSION para poder iniciar sesión en Oracle. El usuario que conceda estos privilegios ha de ser el propietario de las tablas EMPLE y DEPART (o debe tener autorización para conceder privilegios sobre estas tablas) y debe tener el privilegio CREATE SESSION con la posibilidad de concedérselo a otros usuarios:

```
GRANT SELECT, INSERT ON EMPLE TO ACCESO;
```

```
GRANT INSERT ON DEPART TO ACCESO;
```

```
GRANT CREATE SESSION TO ACCESO;
```

Para conceder el rol a un usuario escribimos: GRANT ACCESO TO USUARIO; El usuario podrá conectarse a la base de datos y hacer SELECT e INSERT en la tabla EMPLE, e INSERT en la tabla DEPART.

Se pueden añadir privilegios al rol ejecutando otra orden GRANT. Por ejemplo, añadimos el privilegio SELECT sobre la tabla DEPART al rol creado anteriormente: GRANT SELECT ON DEPART TO ACCESO;

Actividades propuestas



- 3 Crea un usuario y concédele el rol creado (ACCESO). Añade el privilegio CREATE TABLE al rol. Consulta los privilegios de sistema que tiene asignados el usuario creado.



12. Administración de Oracle I

12.5 Gestión de seguridad

- **Límites en privilegios sobre roles**

Un rol puede decidir el acceso de un usuario a un objeto, pero no puede permitir la creación de objetos. Supongamos que el usuario creado anteriormente puede crear vistas (tiene el privilegio CREATE VIEW). También puede hacer SELECT en la tabla EMPLE, ya que tiene asignado el rol ACCESO. Pero no puede crear una vista sobre la tabla EMPLE debido a que recibió el privilegio SELECT a través del rol ACCESO.

- **Supresión de privilegios en los roles**

La orden REVOKE permite suprimir los privilegios dados a los roles. Por ejemplo, para retirar del rol ACCESO el privilegio INSERT en la tabla EMPLE escribiremos lo siguiente: **REVOKE INSERT ON EMPLE FROM ACCESO;** Ahora retiramos del rol ACCESO el privilegio CREATE TABLE: **REVOKE CREATE TABLE FROM ACCESO;**

- **Supresión de un rol**

La orden DROP ROLE permite eliminar un rol de la base de datos. Oracle retira el rol concedido a todos los usuarios y roles a los que se les concedió. Para poder eliminar un rol es necesario ser administrador o tener el privilegio DROP ANY ROLE. Éste es el formato:

DROP ROLE NombreRol;

Por ejemplo, para eliminar el rol ACCESO escribimos: **DROP ROLE ACCESO;**

- **Establecer un rol por defecto**

Es posible establecer un rol por defecto a un usuario mediante la orden ALTER USER, antes tiene que ser concedido el rol al usuario. El formato es:

ALTER USER Nombreusuario DEFAULT ROLE nombre_rol;

Por ejemplo, establecemos un rol por defecto (ACCESO) para el usuario creado anteriormente: **ALTER USER USUARIO DEFAULT ROLE ACCESO;**

- **Información sobre roles en el diccionario de datos**

Para saber a qué usuarios se ha concedido acceso a un rol, o los privilegios que se han concedido a un rol, se pueden consultar las siguientes vistas del diccionario de datos:

- ROLE_SYS_PRIVS: privilegios del sistema asignados a roles.
- ROLE_TAB_PRIVS: privilegios sobre tablas aplicados a roles.
- ROLE_ROLE_PRIVS: roles asignados a otros roles.
- SESSION_ROLES: roles activos para el usuario.
- USER_ROLE_PRIVS: roles asignados al usuario.

Con CREATE USER no se puede asignar un rol por defecto.



D. Perfiles

Un **perfil** es un conjunto de límites a los recursos de la base de datos. Se pueden utilizar perfiles para poner límites a la cantidad de recursos del sistema y de la base de datos disponibles para un usuario y para gestionar las restricciones de contraseña.

Si no se crean perfiles en una base de datos, entonces se utiliza el perfil por defecto (DEFAULT) que especifica recursos ilimitados para todos los usuarios.

La sentencia SQL para crear un perfil es CREATE PROFILE, cuyo formato es:

```
CREATE PROFILE nombreperfil LIMIT
{parámetros_recursos | parámetros_contraseña}
{Entero [K|M] | UNLIMITED | DEFAULT};
```

Donde:

- **parámetros_recursos:** SESSIONS_PER_USER, CPU_PER_SESSION, CPU_PER_CALL, CONNECT_TIME, IDLE_TIME, LOGICAL_READS_PER_SESSION, LOGICAL_READS_PER_CALL, PRIVATE_SGA, COMPOSITE_LIMIT
- **parámetros_contraseña:** FAILED_LOGIN_ATTEMPTS, PASSWORD_LIFE_TIME, PASSWORD_REUSE_TIME, PASSWORD_REUSE_MAX, PASSWORD_LOCK_TIME, PASSWORD_GRACE_TIME, PASSWORD_VERIFY_FUNCTION
- **UNLIMITED** significa que no hay límite sobre un recurso particular.
- **DEFAULT** coge el límite del perfil DEFAULT.

Donde los recursos son los que aparecen en la tabla siguiente:

RECURSO	FUNCIÓN
SESSIONS_PER_USER	Número de sesiones múltiples concurrentes permitidas por nombre de usuario.
CONNECT_TIME	Indica el número de minutos que puede estar una sesión conectada.
IDLE_TIME	Indica el número de minutos que puede estar una sesión conectada sin ser utilizada de forma activa.
CPU_PER_SESSION	Limita el tiempo máximo de CPU por sesión. Este valor se expresa en centésimas de segundo.
CPU_PER_CALL	Limita el tiempo máximo de CPU por llamada (de análisis, ejecución o búsqueda). Se expresa en centésimas de segundo.
LOGICAL_READS_PER_SESSION	Limita el número de bloques de datos leídos en una sesión.
LOGICAL_READS_PER_CALL	Limita el número de bloques de datos leídos por llamada (de análisis, ejecución o búsqueda).

(Continúa)



12. Administración de Oracle I

12.5 Gestión de seguridad

(Continuación)

RECURSO	FUNCIÓN
PRIVATE_SGA	Indica la cantidad de espacio privado que una sesión puede reservar en el área SQL compartida de la SGA (para la opción servidor compartido).
COMPOSITE_LIMIT	Indica un límite compuesto basado en los límites anteriores.
FAILED_LOGIN_ATTEMPTS	Número de intentos de acceso sin éxito consecutivos que producirá el bloqueo de la cuenta.
PASSWORD_LIFE_TIME	Número de días que puede utilizarse una contraseña antes de que caduque.
PASSWORD_REUSE_TIME	Número de días que deben pasar antes de que se pueda reutilizar una contraseña.
PASSWORD_REUSE_MAX	Número de veces que debe cambiarse una contraseña antes de poder reutilizarla.
PASSWORD_LOCK_TIME	Número de días que quedará bloqueada una cuenta si se sobrepasa el valor del parámetro FAILED_LOGIN_ATTEMPTS.
PASSWORD_GRACE_TIME	La duración en días del periodo de gracia durante el cual una contraseña puede cambiarse cuando ha alcanzado su valor PASSWORD_LIFE_TIME.

Para activar el uso de perfiles en el sistema, el administrador ha de ejecutar esta orden: **ALTER SYSTEM SET RESOURCE_LIMIT=TRUE;** (FALSE desactiva la utilización de perfiles). Para asignar un perfil a un usuario se puede utilizar la orden **ALTER USER: ALTER USER USUARIO PROFILE nombreperfil;** O bien al crear el usuario se le puede asignar un perfil.



Caso práctico

- 8 Por razones de seguridad, creamos el perfil **PERFIL1**, en el que limitamos a uno el número de sesiones concurrentes por usuario y a dos minutos el tiempo de conexión permitido por sesión:

```
CREATE PROFILE PERFIL1 LIMIT SESSIONS_PER_USER 1 CONNECT_TIME 2;
```

Dado que los demás límites de recurso no se mencionan en la instrucción CREATE PROFILE, se utilizarán los valores asignados por defecto por el sistema.

A continuación, se crea un usuario llamado PRUEBA. Se le asigna este perfil y se le concede el rol CONNECT:

```
CREATE USER PRUEBA IDENTIFIED BY PRUEBAQUOTA 100K ON USERS PROFILE PERFIL1;
GRANT CONNECT TO PRUEBA;
```

(Continúa)



(Continuación)

Ejecutamos SQL*Plus una vez con el usuario PRUEBA y se conecta perfectamente. Si volvemos a ejecutar SQL*Plus (sin cerrar la sesión anterior) con el usuario PRUEBA, nos aparecerá este mensaje de error:

ERROR:
ORA-02391: exceeded simultaneous SESSIONS_PER_USER limit

Esto se debe a que PRUEBA tiene limitado el número de sesiones concurrentes a una.

Al cabo de un tiempo conectado, el usuario PRUEBA intenta acceder a una tabla, y Oracle visualiza un mensaje de error, indicando que se ha pasado el tiempo de conexión que tenía asignado:

ERROR:
ORA-02399: ha excedido el tiempo máximo de conexión, desconectando

A continuación, se crea un perfil en el que se permiten 3 intentos de acceso fallidos para la cuenta, el cuarto producirá el bloqueo de la cuenta: **CREATE PROFILE PERFIL2 LIMIT FAILED_LOGIN_ATTEMPTS 3;**

Se asigna el perfil al usuario PRUEBA: **ALTER USER PRUEBA PROFILE PERFIL2;** Si hay 3 conexiones fallidas consecutivas en la cuenta PRUEBA (por ejemplo, se ha escrito mal la contraseña) quedará automáticamente bloqueada por Oracle, aunque se conecte con la contraseña correcta:

SQL> CONNECT PRUEBA/PRUEBA
ERROR:
ORA-28000: the account is locked

Para desbloquear una cuenta el administrador de la base de datos tiene que ejecutar: **ALTER USER PRUEBA ACCOUNT UNLOCK;**

A continuación, se modifica el perfil anterior que obligará a los usuarios que lo tengan asignado a cambiar su contraseña cada 2 días:

ALTER PROFILE PERFIL2 LIMIT PASSWORD_LIFE_TIME 2;

La vista DBA_PROFILES contiene información sobre los límites.

La orden ALTER PROFILE permite modificar una determinada configuración de perfil. El formato es el mismo que el de la orden CREATE PROFILE.

Borrado de un perfil

Para borrar un perfil de la base de datos se usa la orden DROP PROFILE, que tiene el siguiente formato:

DROP PROFILE NombrePerfil [CASCADE];

Si algún usuario lo tiene asignado es necesario incluir la opción CASCADE. Ejemplo: **DROP PROFILE PERFIL2 CASCADE;**



12. Administración de Oracle I

Conceptos básicos

Conceptos básicos



A continuación, se muestra un resumen de las órdenes vistas en la Unidad:

USUARIOS

CREATE USER
DROP USER
ALTER USER

PRIVILEGIOS

GRANT
REVOKE

ROLES

CREATE ROLE
DROP ROLE
ALTER ROLE

PERFILES

CREATE PROFILE
DROP PROFILE
ALTER PROFILE

TABLESPACES

CREATE TABLESPACE
DROP TABLESPACE
ALTER TABLESPACE

SECUENCIAS

CREATE SEQUENCE
DROP SEQUENCE
ALTER SEQUENCE

ÍNDICES

CREATE INDEX
DROP INDEX
ALTER INDEX

DATABASE LINKS

CREATE DATABASE LINK
DROP DATABASE LINK
ALTER DATABASE LINK



Actividades complementarias



- 1** Crea una base de datos de forma manual llamada ORACLE10. Sigue el procedimiento del Caso práctico 2.
- 2** Una vez creada la base de datos, crea un usuario y dale permiso para conectarse y crear objetos. Crea una tabla con el formato que deseas. Resuelve las situaciones que se presenten.
- 3** Crea un rol que tenga los siguientes privilegios: INSERT y SELECT en DEPART y EMPLE, CREATE SESSION, CREATE TYPE, CREATE TABLE y CREATE VIEW.
- 4** Crea un usuario llamado COMPRADOR. El *tablespace*, por defecto, es COMPRAS. Se le asigna 1 Megabyte en el *tablespace* COMPRAS. El *tablespace* temporal será TEMP. Se le asigna el rol anterior.
- 5** Realiza la siguiente secuencia de instrucciones en el orden indicado:
1. Crea un usuario de base de datos que tenga funciones de administrador.
 2. Conéctate con el nombre de usuario creado.
 3. Crea varias tablas en el propio esquema.
 4. Crea cinco usuarios nuevos asignándoles un *tablespace* por defecto y cuota (USU1, USU2, USU3, USU4 y USU5).
 5. Da permiso a uno de los usuarios (USU1) sólo para que pueda conectarse a la base de datos.
 6. Crea un rol que permita conectarse a la base de datos y hacer SELECT sobre algunas tablas.
 7. Concede el rol creado a dos de los usuarios creados anteriormente (USU2 y USU3).
 8. Concede al usuario USU4 privilegios sobre algunas tablas con la opción de poder concedérselos a otros usuarios.
- 9** Concede al usuario USU5 cuatro privilegios de sistema, dos de ellos, con la opción de poder concedérselos a otros usuarios.
- 10** Concede a todos los usuarios de la base de datos privilegios para que puedan modificar ciertas columnas de algunas tablas.
- 11** Quita a los usuarios USU3 y USU4 todos los privilegios que tenían asignados.
- 12** Haz que USU5 sólo pueda conectarse en dos sesiones concurrentes a la vez.
- 13** Limita el tiempo de conexión a la base de datos a cinco minutos a los usuarios USU2 y USU3.
- 6** Indica cuál de las siguientes afirmaciones es correcta:
- a) A nivel lógico, una Base de Datos se divide en espacios de tabla (*tablespaces*), segmentos, extensiones y bloques de Oracle.
 - b) Un segmento es una agrupación de bloques de Oracle contiguos.
 - c) Los archivos que componen una base de datos constituyen una división lógica de la misma.
 - d) Un *tablespace* se almacena en un único archivo físico.
- 7** Despues de crear el usuario pepe:
- ```
CREATE USER PEPE IDENTIFIED BY PEPE
DEFAULT TABLESPACE DATOS
QUOTA 1M ON DATOS2;
```
- Indica la o las afirmaciones correctas:
- a) Pepe puede crear objetos en el *tablespace* DATOS.
  - b) Pepe puede crear objetos en el *tablespace* DATOS2.
  - c) Pepe no puede crear objetos en el *tablespace* DATOS porque no tiene espacio asignado en dicho *tablespace*.
  - d) Pepe puede crear objetos en los dos *tablespaces*.

# Administración de Oracle II

13

R

En esta unidad aprenderás a:

- 1 Crear y gestionar los tablespaces.
- 2 Utilizar secuencias e índices.
- 3 Crear copias de seguridad.
- 4 Restaurar una base de datos.
- 5 Analizar los registros de deshacer o Redo Log.
- 6 Realizar auditorías a objetos de bases de datos.



## 13.1 Gestión de tablespaces

Ya se ha explicado que una base de datos está formada por un conjunto de archivos de datos, pero ¿cómo agrupa Oracle estos archivos? Lo hace usando un objeto denominado *tablespace* o *espacio de tablas*. Se llama así porque contiene tablas de datos. Antes de introducir datos en la base de datos, es necesario crear un *tablespace* y, seguidamente, las tablas en las que se van a introducir los datos. Estas tablas se deben almacenar en un *tablespace*.

Un **tablespace** es una unidad lógica de almacenamiento de datos representada físicamente por uno o más archivos de datos. Se recomienda no mezclar datos de diferentes aplicaciones en el mismo *tablespace*, es decir, se debe crear un *tablespace* para almacenar los datos de la aplicación de gestión de almacén, otro *tablespace* para almacenar los datos de la nómina de los empleados, etcétera. Al instalar Oracle se crean varios: SYSTEM, USERS, TEMP o UNDOTBS1. Veamos a continuación cómo se crean y gestionan los *tablespaces*.

### A. Creación de un tablespace

Para crear un *tablespace* se usa la orden CREATE TABLESPACE que permite asignar uno o más archivos al espacio de tablas y especificar un espacio por omisión para cualquiera de las tablas creadas sin un espacio de tabla explícitamente mencionado en una sentencia CREATE TABLE. El formato es:

```
CREATE [UNDO] TABLESPACE nombretablespace
DATAFILE 'nombrearchivo' [SIZE entero[K|M]] [REUSE]
 [AUTOEXTEND {OFF|ON cláusulas}]
 [, 'nombrearchivo' [SIZE entero[K|M]] [REUSE]
 [AUTOEXTEND {OFF|ON cláusulas}]] ...
 [DEFAULT STORAGE
 (
 INITIAL tamaño
 NEXT tamaño
 MINEXTENTS tamaño
 MAXEXTENTS tamaño
 PCTINCREASE valor
)]
 [ONLINE|OFFLINE]
 [PERMANENT|TEMPORARY]
 [EXTENT MANAGEMENT
 {DICTIONARY|LOCAL {AUTOALLOCATE |UNIFORM [SIZE tamaño
 [K|M]]}}}] ;
```

Donde:

- UNDO especifica que se crea un *tablespace* de tipo «deshacer cambios», es decir, de Rollback. Si no hay un *tablespace* de deshacer se crea uno con la opción UNDO y Oracle lo asigna automáticamente a la instancia como el *tablespace* de deshacer. En éste no se podrán crear objetos. Está reservado para las acciones de deshacer cambios.

Un **bloque de datos** es la unidad de acceso a disco de Oracle.

Una **extensión** es un conjunto de bloques de datos contiguos.

Un **segmento** es un conjunto de extensiones.



## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces

- **DATAFILE** especifica el archivo o archivos de datos de que constará el *tablespace*.
- **SIZE** entero especifica el tamaño del *tablespace*, que puede venir dado en Kilobytes (K) o en Megabytes (M). Si ponemos K, se multiplica el entero por 1.024. Si ponemos M, se multiplica por 1.048.576.
- **REUSE** reutiliza el archivo si ya existe, o lo crea si no existe.
- **DEFAULT STORAGE** define el almacenamiento por omisión para todos los objetos que se creen en este espacio de tabla. Fija la cantidad de espacio si no se especifica en la sentencia CREATE TABLE.
- **INITIAL** extensión inicial. Especifica el tamaño en bytes de la primera extensión del objeto. El tamaño se puede especificar en Kilobytes (K) o Megabytes (M).
- **NEXT** extensión siguiente. Especifica el tamaño de la siguiente extensión que se va a asignar al objeto. También se puede especificar K o M. El valor por defecto es el tamaño de un bloque de datos (el tamaño del bloque se especifica en un parámetro de inicio de Oracle: *db\_block\_size* y debe ser múltiplo del tamaño del bloque del sistema operativo del servidor).
- **MINEXTENTS** reserva extensiones adicionales más allá de la extensión inicial que se da a la tabla por omisión. Este parámetro permite asignar una gran cantidad de espacio cuando se crea un objeto, incluso si el espacio disponible no está contiguo. El valor por omisión es 1, que significa que Oracle sólo asigna la extensión inicial. Si el valor es mayor que 1, Oracle calcula el tamaño de las extensiones subsiguientes basándose en los valores de los parámetros INITIAL, NEXT y PCTINCREASE.
- **MAXEXTENTS** es el número total de extensiones, incluida la primera, que Oracle puede asignar al objeto. El valor depende del tamaño del bloque de datos.
- **PCTINCREASE** es un factor de crecimiento para la extensión. El valor por omisión es 50, lo que significa que cada extensión subsiguiente será un 50 por 100 más grande que la extensión anterior. El valor de la siguiente extensión es:  $\text{NEXT} = \text{NEXT} + (\text{PCTINCREASE} * \text{NEXT}) / 100$
- **ONLINE, OFFLINE**. Con **ONLINE** el *tablespace* está disponible después de crearlo, es el valor por defecto. **OFFLINE** impide su acceso.
- **PERMANENT, TEMPORARY**. **PERMANENT** es la opción por defecto. **TEMPORARY** indica que el *tablespace* solo podrá albergar objetos temporales. Por ejemplo, segmentos que se utilizan en ordenaciones de datos.
- **AUTOEXTEND** cláusulas: activa o desactiva el crecimiento automático de los archivos de datos del *tablespace*. Cuando un *tablespace* se llena podemos usar esta opción para que el tamaño del archivo o archivos de datos asociados crezca automáticamente. **AUTOEXTEND OFF** desactiva el crecimiento automático. El formato **AUTOEXTEND ON** es:

```
AUTOEXTEND ON NEXT entero {K|M} MAXSIZE {UNLIMITED|entero {K|M}}
```

## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces



**NEXT entero:** es el incremento de espacio en disco expresado, en Kilobytes o en Megabytes, que se reservará automáticamente para el archivo.

**MAXSIZE:** es el máximo espacio en disco reservado para la extensión automática del archivo.

**UNLIMITED:** significa que no hay límite del espacio en disco reservado.

- **EXTENT MANAGEMENT** especifica la gestión de las extensiones del *tablespace*:

**DICTIONARY** especifica que se gestiona mediante tablas de diccionario. Es la opción por defecto si el parámetro **COMPATIBLE** es inferior a 9.0.0.

**LOCAL** especifica que se gestiona localmente mediante un mapa de bits.

**AUTOALLOCATE:** el *tablespace* lo gestiona el sistema. Los usuarios no podrán especificar el tamaño de una extensión. Es la opción por defecto si el parámetro **COMPATIBLE** es 9.0.0 o superior.

**UNIFORM** especifica que el *tablespace* se gestiona con extensiones uniformes de bytes **SIZE**. El tamaño **SIZE** por defecto es 1 MB. Si se especifica **LOCAL** no se puede especificar **DEFAULT**, **MINIMUM EXTENT** o **TEMPORARY**.

### Caso práctico

- 1 A continuación, se crea un *tablespace* de 15 Megabytes llamado TRABAJO. El tamaño inicial para el objeto que se cree en el *tablespace* (por ejemplo, una tabla) es de 10 K. El tamaño de la siguiente extensión del objeto también es 10 K; cada extensión subsiguiente será un 25 por 100 más grande que la anterior. Asignamos dos archivos a este *tablespace* 'TRABAJO1.ORA', de 10M, y 'TRABAJO2.ORA', de 5M: (si no ponemos la ubicación el *tablespace* lo creará en C:\WINNT\SYSTEM32):

```
CREATE TABLESPACE TRABAJO DATAFILE 'TRABAJO1.ORA' SIZE 10M,
 'TRABAJO2.ORA' SIZE 5M
 DEFAULT STORAGE (INITIAL 10K NEXT 10K PCTINCREASE 25);
```

Se crea un *tablespace* de 100K llamado PEQUE. Asignamos el archivo 'PEQUE.ORA', habilitando el crecimiento automático de 120K para la extensión siguiente dentro de un espacio máximo de 1M:

```
CREATE TABLESPACE PEQUE DATAFILE 'PEQUE.ORA' SIZE 100K
 AUTOEXTEND ON NEXT 120K MAXSIZE 1M;
```

Se crea un *tablespace* de deshacer de 10M llamado DESHACER. Asignamos el archivo 'DESHACER.ORA', habilitando el crecimiento automático de 512K para la extensión siguiente dentro de un espacio máximo de ilimitado:

```
CREATE UNDO TABLESPACE DESHACER DATAFILE 'DESHACER.ORA' SIZE 10M REUSE AUTOEXTEND ON NEXT 512K MAXSIZE UNLIMITED;
```



## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces

- **Tablespaces temporales**

Aunque la creación de un *tablespace* temporal se puede hacer con el comando anterior, es aconsejable utilizar la orden CREATE TEMPORARY TABLESPACE. El formato es el que sigue:

```
CREATE TEMPORARY TABLESPACE nombretablespace
TEMPFILE 'nombrearchivo' [SIZE entero[K|M]] [REUSE]
[AUTOEXTEND {OFF|ON cláusulas}]
[, 'nombrearchivo' [SIZE entero[K|M]] [REUSE]
[AUTOEXTEND {OFF|ON cláusulas}]]...
[MAXSIZE {UNLIMITED | entero {K|M}}]
[EXTENT MANAGEMENT LOCAL UNIFORM [SIZE entero
{K|M}]];
```

TEMPFILE se puede omitir si se ha definido el parámetro de inicialización DB\_CREATE\_FILE\_DEST, en el que Oracle crea un archivo temporal de 100 MB gestionado por él mismo.

El tamaño de las extensiones SIZE serán múltiplos del parámetro SORT\_AREA\_SIZE. El segmento temporal recibirá bloques del tamaño que aquí pongamos. Por ejemplo, la orden que se muestra crea un *tablespace* temporal de 10 M llamado TEMPEJER y se ubicará en el TEMPFILE con el nombre indicado:

```
CREATE TEMPORARY TABLESPACE TEMPEJER TEMPFILE 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\TEMP_EJERCI.DBF' SIZE
10M;
```

- **Vistas con información sobre tablespaces**

Existen varias vistas para obtener información sobre *tablespaces*. Algunas de ellas son:

- **DBA\_DATA\_FILES:** Muestra información sobre los archivos utilizados por los *tablespaces* (FILE\_NAME). El espacio está definido en bytes (BYTES). El máximo tamaño que puede llegar a tener (MAXBYTES), etc. Para consultarla es necesario que el usuario SYS dé privilegios.
- **USER\_FREE\_SPACE:** Muestra las extensiones libres en *tablespaces* a las que puede acceder el usuario. No tienen por qué estar en bloques consecutivos. Por ejemplo, conectamos como usuario PRUEBA y consultamos las extensiones libres. Las columnas para esta vista son:

FILE\_ID: N°. de identificación del archivo.

BYTES: Es el número bytes libres.

BLOCK\_ID: Identificación del primer bloque libre.

BLOCKS: N°. de bloques libres.

RELATIVE\_FNO: Número relativo del fichero en la primera extensión del bloque.

- **DBA\_FREE\_SPACE:** Muestra extensiones libres en todos los *tablespaces*.

## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces



- **DBA\_TABLESPACES:** Muestra la descripción de todos los *tablespaces*. La orden siguiente visualiza los nombres de los *tablespaces* y el tipo:

```
SQL> SELECT TABLESPACE_NAME, CONTENTS FROM DBA_TABLESPACES;
TABLESPACE_NAME CONTENTS
----- -----
SYSTEM PERMANENT
UNDOTBS1 UNDO
SYSAUX PERMANENT
TEMP TEMPORARY
USERS PERMANENT
EXAMPLE PERMANENT
TRABAJO PERMANENT
PEQUE PERMANENT
DESHACER UNDO
TEMPEJER TEMPORARY
```

- **DBA\_TS\_QUOTAS:** Muestra los bytes utilizados por los usuarios en cada *tablespace*. Para consultar estas vistas es necesario que el usuario SYS nos dé privilegios. Las columnas para esta vista son:

**BYTES :** Es el número de bytes usados por el usuario.

**MAX\_BYTES :** Nº. máximo de bytes que tiene el usuario asignado. Si el valor es -1 significa que tiene asignado un Nº. de bytes ilimitado.

**BLOCKS :** Nº. de bloques usados.

**MAX\_BLOCKS :** Máximo número de bloques.

#### Actividades propuestas



- 1 A partir de las vistas DBA\_DATA\_FILES y DBA\_TS\_QUOTAS, haz una consulta que contenga la siguiente información: Nombre del tablespace, Nombre Usuario, Tamaño total del tablespace, bytes usados por el usuario.

## B. Modificación de tablespaces

Los *tablespaces*, una vez creados, se pueden modificar, es decir, es posible añadir nuevos archivos a un *tablespace* existente, modificar las cláusulas de almacenamiento para los objetos que se almacenen en el *tablespace*, activarlo y desactivarlo, etcétera. La modificación se lleva a cabo con la orden ALTER TABLESPACE, cuyo formato es:



## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces

```
ALTER TABLESPACE nombretablespace
{
 [ADD DATAFILE 'nombrearchivo' [SIZE entero[K|M]] [REUSE]
 [AUTOEXTEND ON ..|OFF]
 [, 'nombrearchivo' [SIZE ntero[K|M]] [REUSE]
 [AUTOEXTEND ON ..|OFF]]..
]
 [RENAME DATAFILE 'archivo' [, 'archivo']...
 TO 'archivo' [, 'archivo']]]
 [DEFAULT STORAGE ClasulasAlmacenamiento]
 [ONLINE|OFFLINE]
} ;
```

Donde:

- *nombretablespace* es el nombre del *tablespace* que se quiere modificar.
- *ADD\_DATAFILE* añade al *tablespace* uno o varios archivos.
- *RENAME DATAFILE* cambia el nombre de un archivo existente del *tablespace*. Este cambio se tiene que hacer desde el sistema operativo y, después, ejecutar la orden SQL. El *tablespace* debe estar desactivado (*offline*) mientras se produce el cambio.
- *DEFAULT STORAGE* especifica los nuevos parámetros de almacenamiento para todos los objetos que se creen a partir de ahora en este *tablespace*.
- *ONLINE* pone el espacio de tablas en línea (activado).
- *OFFLINE* pone el espacio de tablas fuera de línea (desactivado).



### Caso práctico

#### 2 Se agrega un archivo al *tablespace* TRABAJO de 6 Megabytes llamado 'TRABAJO3.ORA'

```
ALTER TABLESPACE TRABAJO ADD DATAFILE 'TRABAJO3.ORA' SIZE 6M;
```

Renombramos los archivos TRABAJO1.ORA y TRABAJO2.ORA del *tablespace* TRABAJO se llamarán ahora TRABA1.ORA y TRABA2.ORA, respectivamente. Los pasos para renombrar archivos asociados a un *tablespace* son los siguientes:

1. Desactivar el *tablespace* TRABAJO: **ALTER TABLESPACE TRABAJO OFFLINE;**
2. Copiar los archivos TRABAJO1.ORA y TRABAJO2.ORA a TRABA1.ORA y TRABA2.ORA utilizando los comandos del sistema operativo.
3. Usar la orden ALTER TABLESPACE con la opción RENAME DATAFILE: **ALTER TABLESPACE TRABAJO RENAME DATAFILE 'TRABAJO1.ORA', 'TRABAJO2.ORA' TO 'TRABA1.ORA', 'TRABA2.ORA';**
4. Activar el *tablespace* TRABAJO: **ALTER TABLESPACE TRABAJO ONLINE;**



#### Actividades propuestas



2 Vamos a probar la opción de crecimiento automático para los archivos de un *tablespace*. Partimos del *tablespace* PEQUE creado anteriormente. Crea una tabla en este *tablespace* e inserta filas hasta que Oracle devuelva un error (porque se ha llenado el *tablespace*).

- Modifica entonces el *tablespace* PEQUE añadiendo un archivo llamado 'MAYOR.ORA' de 100 K, que se va extendiendo automáticamente hasta 200 K cuando se llena. Como máximo, el espacio utilizado en disco para este archivo será de 1 MB.
- Sigue insertando filas hasta que sobrepase 1 MB de espacio en disco. Oracle visualizará un mensaje de error similar al anterior. Modificalo de nuevo añadiendo un archivo de 100 K, que se va extendiendo automáticamente a 200 K sin indicar máximo tamaño. Prueba de nuevo a insertar filas.

### C. Borrado de tablespaces

Para borrar un *tablespace* que ya no utilizamos se emplea **DROP TABLESPACE**.

Su formato es:

```
DROP TABLESPACE nombretablespace
[INCLUDING CONTENTS [AND DATAFILES] [CASCADE CONSTRAINTS]] ;
```

Donde:

- nombretablespace es el nombre del *tablespace* que se va a suprimir.
- La opción **INCLUDING CONTENTS** permite borrar un *tablespace* que tenga datos. Sin esta opción, únicamente se puede suprimir un *tablespace* vacío.
- **AND DATAFILES** borra los archivos de datos asociados y **CASCADE CONSTRAINTS** borra las relaciones de integridad referencial que afecten a las tablas del *tablespace* suprimido.

Las órdenes **CREATE TABLE** y **CREATE TABLESPACE** contienen más cláusulas. En el tema sólo se han visto algunas. Para más información consultar el manual de referencia de Oracle.

Se recomienda poner el *tablespace* OFFLINE antes de borrarlo para asegurarnos de que no haya sentencias SQL que estén accediendo a datos del *tablespace*, en cuyo caso no sería posible borrarlo. Por ejemplo, para borrar el *tablespace* PEQUE y los archivos de datos asociados escribiremos:

```
DROP TABLESPACE PEQUE INCLUDING CONTENTS AND DATAFILES ;
```



## 13. Administración de Oracle II

### 13.1 Gestión de tablespaces

#### D. Parámetros de almacenamiento

Hasta ahora, en las tablas que hemos creado no se ha definido ningún parámetro de almacenamiento, pues éste venía definido en la cláusula DEFAULT STORAGE del *tablespace* que contiene a la tabla. No obstante, en la orden CREATE TABLE podemos especificar los parámetros de almacenamiento para la tabla cuando tengamos que asignar parámetros distintos a los que asigna el *tablespace*.

El formato de CREATE TABLE, usando estos parámetros, es el siguiente:

```
CREATE TABLE Nombretabla
(
 Columna1 Tipo_dato [NOT NULL],
 Columna2 Tipo_dato [NOT NULL],
 [restricciones_de_tabla]
)
STORAGE
(
 INITIAL tamaño
 NEXT tamaño
 MINEXTENTS tamaño
 MAXEXTENTS tamaño
 PCTINCREASE valor
)
[TABLESPACE nombretablespace];
```

El significado de los parámetros de la cláusula STORAGE es el mismo que el de los parámetros usados en la orden CREATE TABLESPACE.

#### E. Desconexión de un tablespace

Con la base de datos abierta se pueden desconectar los *tablespaces*, excepto SYSTEM o cualquier *tablespace* con algún segmento de Rollback activo o segmentos temporales. Cuando se desconecta un *tablespace* el servidor Oracle desconecta los archivos asociados. Los usuarios que intenten acceder a objetos de un *tablespace* desconectado obtendrán un mensaje de error. El formato es el siguiente:

```
ALTER TABLESPACE nombretablespace { ONLINE |
OFFLINE [NORMAL|TEMPORARY|IMMEDIATE] };
```

OFFLINE. Desactiva el *tablespace* e impide el acceso a sus segmentos. En modo NORMAL limpia la SGA de todos los bloques de los archivos de datos de este *tablespace*. Es la opción predeterminada. No es necesario recuperarlo antes de volver a activarlo.

Con TEMPORARY, Oracle ejecuta un *checkpoint* para todos los archivos de datos activos del *tablespace*. Antes de activarlo puede requerir recuperación. IMMEDIATE no ejecuta *checkpoint* y no asegura la disponibilidad de los archivos de datos asociados. Antes de activarlo requiere recuperación física.



## 13.2 Otros objetos

En este apartado se estudian otros objetos que se han nombrado a lo largo de esta Unidad o en las anteriores y que pueden ser de bastante utilidad: las *secuencias* y los *índices*.

### A. Secuencias

Una **secuencia** es un objeto de base de datos que sirve para generar números enteros únicos. Es muy útil para generar automáticamente valores para claves primarias. Para crear una secuencia en el esquema propio es necesario tener el privilegio CREATE SEQUENCE. Se crea una secuencia en cualquier otro esquema con el privilegio CREATE ANY SEQUENCE. El formato para crear una secuencia es éste:

```
CREATE SEQUENCE nombresecuencia
[INCREMENT BY entero]
[START WITH entero]
[MAXVALUE entero | NOMAXVALUE]
[MINVALUE entero | Nominvalue]
[CYCLE | NOCYCLE]
[ORDER | NOORDER]
[CACHE entero | NOCACHE];
```

Donde:

- INCREMENT BY entero especifica el intervalo de crecimiento de la secuencia. Si se omite, se asume valor 1. Si es negativo, produce un decremento de la secuencia.
- START WITH entero es el número con el que comienza la secuencia.
- MAXVALUE entero es el número más alto que generará la secuencia. Este entero debe ser menor o igual que el entero especificado en START WITH y mayor que el entero especificado en MINVALUE.
- NOMAXVALUE señala que el valor máximo para una secuencia ascendente es  $10^{27}$  y para una secuencia descendente -1.
- MINVALUE entero es el número más bajo que generará la secuencia. El entero debe ser menor o igual que el entero especificado en START WITH y menor que el entero especificado en MAXVALUE.
- Nominvalue indica que el valor mínimo para una secuencia ascendente es 1 y  $-10^{26}$  para una secuencia descendente.
- CYCLE | NOCYCLE. CYCLE reanuda la secuencia cuando llega al máximo o al mínimo valor; NOCYCLE no la reanuda.
- ORDER | NOORDER. ORDER garantiza que los números de secuencia se generan en el orden requerido; NOORDER no lo garantiza. Si se omiten ambas, se asume NOORDER. En modo exclusivo, las secuencias siempre se generan en orden ascendente.

Los valores por defecto son: INCREMENT BY 1, NOMINVALUE, NOMAXVALUE y NOORDER.



## 13. Administración de Oracle II

### 13.2 Otros objetos

- CACHE entero |NOCACHE. CACHE permite guardar en memoria un conjunto previamente asignado de números de secuencia para garantizar acceso más rápido. En secuencias cíclicas, este valor debe ser menor que el número de valores del ciclo. El mínimo valor es 2. NOCACHE indica que los valores de la secuencia no se pueden precalcular.

Una vez creada la secuencia, accedemos a ella mediante las pseudocolumnas CURRVAL, que devuelve el valor actual de la secuencia, y NEXTVAL, que devuelve el siguiente valor e incrementa la secuencia. Para acceder a estos valores tenemos que poner el nombre de la secuencia, un punto y, a continuación, la pseudocolumna: NOMBRESEQUENCIA.CURRVAL NOMBRESEQUENCIA.NEXTVAL.



### Caso práctico

- 3 Se crea una tabla llamada FRUTAS con dos columnas: CODIGO y NOMBRE. La columna CODIGO se define como clave primaria:

```
CREATE TABLE FRUTAS (CODIGO NUMBER(2) NOT NULL PRIMARY KEY, NOMBRE VARCHAR2(15));
```

Ahora se crea una secuencia llamada CODIGOS que generará números empezando por el valor 1, con incremento 1 y cuyo máximo valor para la secuencia será 99: `CREATE SEQUENCE CODIGOS START WITH 1 INCREMENT BY 1 MAXVALUE 99;`

Se insertan filas en la tabla FRUTAS usando la secuencia CODIGOS para generar el CODIGO de cada fila de la tabla:

```
INSERT INTO FRUTAS VALUES (CODIGOS.NEXTVAL, 'MANZANAS');
INSERT INTO FRUTAS VALUES (CODIGOS.NEXTVAL, 'NARANJAS');
INSERT INTO FRUTAS VALUES (CODIGOS.NEXTVAL, 'PERAS');
```

Para consultar el valor actual de la secuencia escribimos: `SELECT CODIGOS.CURRVAL FROM DUAL;`



### Actividades propuestas

- 3 Crea una secuencia cíclica que comience en 1, se incremente en 1, y el máximo valor sea de 10. Usa la secuencia y comprueba que al llegar a 10 vuelve a iniciarse.

Para eliminar una secuencia de la base se usa la orden `DROP SEQUENCE`. Por ejemplo, borramos la secuencia creada anteriormente: `DROP SEQUENCE CODIGOS;`



## B. Índices

Con los índices se acelera el tiempo de respuesta en las consultas. Un **índice** es un objeto de base de datos que se asocia a una tabla y al que se asocia una o varias columnas de la tabla. Se puede almacenar en un *tablespace* diferente al de la tabla que indexa. Contienen un elemento para cada valor que aparece en la columna o columnas indexadas de la tabla, y proporciona un acceso rápido y directo a las filas mediante el ROWID.

Estudiaremos la utilidad de los índices con un ejemplo: Se supone que Hacienda dispone de una tabla de CONTRIBUYENTES en la que cada fila se identifica por el NIF del contribuyente. Para consultar un NIF determinado realizamos la siguiente consulta: `SELECT * FROM CONTRIBUYENTES WHERE NIF='7866978-A';`

Si la columna NIF no está indexada, Oracle recorre la tabla de CONTRIBUYENTES secuencialmente, desde la primera fila hasta el final de la tabla, tanto si encuentra como si no encuentra el dato que se busca. Si la columna NIF está indexada, Oracle realizará una búsqueda binaria en el índice hasta encontrar el dato buscado, obteniendo el ROWID de la fila asociada mediante un único acceso sobre la tabla CONTRIBUYENTES.

Se debe indexar cuando se disponga de una gran cantidad de filas en una tabla. Cada fila se debe identificar con una o varias columnas. Es conveniente no indexar las tablas pequeñas, a no ser que sea necesario definir claves primarias o columnas únicas. No se deben indexar columnas que son modificadas (UPDATE) a menudo y que posean pocos valores diferentes. Tampoco es útil indexar tablas en las que se haga una gran cantidad de sentencias UPDATE, DELETE o INSERT, ya que la modificación de la tabla implica la modificación del índice.

El formato para crear un índice es el siguiente:

```
CREATE INDEX nombreindice
ON nombretabla (colum[ASC|DESC] [,colum[ASC|DESC]]...)
[STORAGE clausulas_almacenamiento]
[TABLESPACE nombretablespace]
[otras_cláusulas];
```

Donde:

- *Nombretabla* es la tabla que se va a indexar.
- *colum* es la columna o columnas que se indexan.
- *ASC/DESC* se especifica para definir índices ascendentes o descendentes.

No hay que olvidar que cuando se hace una restricción de clave primaria (PRIMARY KEY) o una restricción de unicidad (UNIQUE) se crea un índice con el nombre de la restricción. Por ejemplo, para crear un índice en la columna EMP\_NO para la tabla EMPLE escribiremos: `CREATE INDEX INDICEEMPLE ON EMPLE (EMP_NO);`

Las vistas USER\_INDEXES y DBA\_INDEXES informan sobre los índices creados.

Para eliminar un índice de la base de datos se usa la orden `DROP INDEX`. Por ejemplo, borramos el índice creado anteriormente: `DROP INDEX INDICEEMPLE;`



## 13. Administración de Oracle II

### 13.2 Otros objetos

#### C. Enlaces de bases de datos

Para explicar los enlaces de bases de datos, vamos a suponer que disponemos en nuestra aula de dos servidores, un Servidor Novell y un Servidor Windows 2003, y que en ambos tenemos instalada la base de datos Oracle. Todos los usuarios tienen cuenta para conectarse a las dos bases de datos. Si nos conectamos desde un puesto a la base de datos que está en el Servidor Windows 2003 y queremos utilizar los datos de las tablas que están en la base de datos instalada en el Servidor Novell, tendremos que crear un enlace de base de datos o **DATABASE LINK**. Un **enlace de base de datos** es un objeto que permite acceder a objetos de una base de datos remota. Define el enlace entre una base de datos local y un nombre de usuario en una base de datos remota. Se usa para realizar consultas en tablas de la base de datos remota.

Para crear un enlace de base de datos se emplea la orden de SQL, **CREATE DATABASE LINK**, cuyo formato es:

```
CREATE [PUBLIC] DATABASE LINK nombredeenlace
CONNECT TO usuario IDENTIFIED BY clave
USING 'cadena_de_conexión';
```

Donde:

- **PUBLIC** crea un enlace de base de datos público.
- **cadena\_de\_conexión** es la cadena de conexión utilizada para conectar con la base de datos remota.
- **usuario/clave** es el nombre de usuario y la contraseña utilizados para conectarse a la base de datos remota. El nombre de usuario tiene que existir en la base de datos remota.

En el acceso a tablas remotas es necesario añadir al nombre de la tabla el nombre del enlace de la siguiente manera: **NOMBRETABLA@NOMBREDEENLACE**.



#### Caso práctico

- 4 Para conectar a la base de datos remota (por ejemplo, está almacenada en un Servidor Novell), cuya cadena de conexión es 'DAI' y el nombre de usuario y su clave es MAJESUS/RAMOS, se crea el siguiente enlace de base de datos:

```
CREATE DATABASE LINK MIENLACE CONNECT TO MAJESUS IDENTIFIED BY RAMOS USING 'DAI';
```

Se puede acceder a la tabla **EMPLE** de MAJESUS de la siguiente manera: **SELECT \* FROM EMPLE@MIENLACE;**

Para eliminar un enlace de base de datos se usa la orden **DROP DATABASE LINK**. Éste es su formato:

```
DROP [PUBLIC] DATABASE LINK nombredeenlace;
```

Por ejemplo, borramos el enlace anterior: **DROP DATABASE LINK MIENLACE;**



## 13.3 Copias de seguridad

Una de las funciones más importantes de un administrador de bases de datos es el mantener los datos siempre disponibles, proteger la base de datos de todos los posibles fallos y hacer que esté operativa lo antes posible ante cualquier fallo. Para ello, el DBA debe planificar y realizar copias de seguridad con regularidad. Las copias de seguridad son esenciales a la hora de recuperar la base de datos ante fallos inesperados.

Oracle ofrece una variedad de procedimientos de copia de seguridad y opciones que facilitan la protección de bases de datos. Puede haber *copias de seguridad lógicas* y *copias físicas*. Existen tres métodos: *exportaciones* o *copias lógicas*, *copias de seguridad fuera de línea* y *en línea*, que son *copias físicas*.

Por norma general, las bases de datos en producción basan su seguridad en copias físicas.

### A. Copias de seguridad lógicas

Las **copias lógicas** consistirán en leer datos de la base de datos y extraerlos generando un archivo de exportación que podrá ser recuperado con la orden IMPORT de Oracle. Se pueden exportar usuarios, tablas y hasta la base de datos completa.

En general, las utilidades IMPORT y EXPORT de Oracle van a permitir:

- Archivar datos históricos. Por ejemplo, hacer copias de los esquemas cada vez que se realiza algún cambio.
- Guardar definiciones de tablas con o sin datos como medida de protección ante fallos de usuarios.
- Migrar datos de una versión a otra de Oracle cuando se actualiza una versión o mover datos entre máquinas con distinto sistema operativo y bases de datos diferentes.
- Transportar *tablespaces* de una base de datos a otra.

### ◆ Utilidad EXPORT

Para ejecutar esta utilidad lo hacemos en modo comando desde el sistema operativo, sin entrar a SQL, PLUS. Para ver el formato y los parámetros del comando EXP escribimos:

```
C:\>exp help=yes
```

Los parámetros más comunes son:

- USERID: Nombre de usuario/contraseña de la cuenta que realiza la exportación. La palabra USERID es opcional.
- BUFFER: Tamaño del buffer de datos en bytes.
- FILE: Nombre del archivo de salida, por defecto es *expdat.dmp*.



## 13. Administración de Oracle II

### 13.3 Copias de seguridad

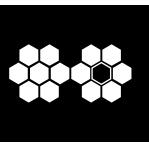
- **COMPRESS:** Indica si la exportación debe comprimir segmentos fragmentados en una misma extensión. Se pone Y o N. Para tablas grandes se utiliza `compress=N`
- **GRANTS:** Especifica si las concesiones sobre los objetos se exportan (Y) o no (N). Por defecto es Y.
- **INDEXES:** Especifica si se exportan los índices de las tablas (Y) o no (N). Por defecto es Y.
- **DIRECT:** Indica si se realiza una exportación directa, sin utilizar la caché. Por defecto es N.
- **LOG:** Nombre del archivo para mensajes informativos y de error.
- **ROWS:** Especifica si se exportan las filas de las tablas (Y) o no (N). Por defecto es Y.
- **CONSISTENT:** Indica si se mantiene una versión con coherencia de lectura de todos los objetos exportados. Necesario cuando las tablas se actualizan mientras se están exportando.
- **FULL:** Se realiza una exportación completa de la base de datos si se pone `FULL=Y`. Por defecto es N.

Si se exportan datos de una instancia a otra hay que asegurarse de que la base de datos destino tenga los mismos nombres de espacios de tablas que la base de datos origen, pues los nombres de espacios de tablas se exportan como parte de la definición de los objetos.

- **OWNER:** Indica una lista de cuentas de la base de datos que se van a exportar.
- **CONSTRAINTS:** Especifica si se exportan las restricciones sobre tablas (Y) o no (N). Por defecto es Y.
- **TRIGGERS:** Especifica si se exportan los triggers (Y) o no (N). Por defecto es Y.
- **TABLES:** Especifica la lista de tablas que se va a exportar.
- **PARFILE:** Especifica el nombre del archivo de parámetros que se va a pasar a EXPORT. Puede contener entradas para todos estos parámetros.
- **QUERY:** Especifica una cláusula WHERE que se aplicará a cada tabla durante la exportación.
- **TABLESPACES:** Especifica los *tablespaces* que se deben transportar.
- **TRANSPORT\_TABLESPACE:** Activa la exportación de *tablespaces* transportables.
- **FEEDBACK:** Visualiza el proceso de exportación cada X filas, por defecto es 0.

A continuación se muestran los modos de exportación de la orden EXPORT:

- **Modo tabla:** en este modo se exportan las definiciones de la o las tablas, los datos o las filas seleccionadas de la tabla, concesiones de tablas del propietario, índices de tablas del propietario y restricciones de tabla.
- **Modo usuario:** se exportan todos los objetos del esquema del usuario, es decir, definiciones de tablas, los datos de las tablas, procedimientos, funciones y paquetes. Concesiones del propietario, índices del propietario y restricciones de tablas.



- **Modo *tablespace*:** esta opción se puede utilizar para mover *tablespaces* de una base de datos a otra. El movimiento de datos mediante *tablespaces* transportables puede ser más rápido que hacer una importación y exportación de los datos puesto que para transportar un *tablespace* sólo es necesario copiar los archivos de datos e integrar su información estructural.
- **Modo base de datos completa:** este modo exporta todos los objetos de la base de datos excepto los del esquema SYS. No coge el diccionario de datos.

### ◆ Utilidad IMPORT

Se usa para la recuperación de archivos de exportación. Se puede utilizar para recuperar objetos o usuarios seleccionados desde el archivo de volcado de exportación. Muchos de los parámetros del comando coinciden con EXPORT. Otros parámetros son:

- **USERID:** Nombre de usuario/contraseña de la cuenta que realiza la importación, la palabra *userid* es opcional, tienen que ser usuarios con privilegios DBA o IMP\_FULL\_DATABASE.
- **FILE:** Nombre del archivo de exportación a importar.
- **SHOW:** Especifica si el contenido del archivo debe mostrarse (S/N) en lugar de ejecutarse. Por defecto es N.
- **IGNORE:** Especifica si la importación debe ignorar errores encontrados (S/N) al ejecutar el comando CREATE. Se utiliza si los objetos a importar ya existen. Por defecto es N.
- **FROMUSER:** Lista de cuentas de la base de datos cuyos objetos deben leerse desde el archivo de exportación cuando FULL=N.
- **TOUSER:** Lista de cuentas de la base de datos en las que se importarán los objetos del archivo de exportación.
- **DESTROY:** Especifica si se ejecutan los comandos CREATE TABLESPACE que se encuentran en los archivos de exportaciones completas, destruyendo los archivos de datos contenidos en la base de datos en la que van a ser importados. Por defecto es N.
- **INDEXFILE:** Esta opción permite escribir todos los comandos CREATE TABLESPACE, CREATE CLUSTER y CREATE INDEX en un archivo en lugar de ejecutarlos. Si CONSTRAINTS=Y, entonces las restricciones se escribirán también en el archivo. Este archivo puede entonces ejecutarse tras importarlo con INDEXES=N, será necesario hacer pequeños cambios.
- **RESUMABLE:** Especifica si se continúa la sesión (S/N) después de los errores. Por defecto es N.
- **COMPILE:** Especifica si se deben compilar (S/N) de nuevo procedimientos, funciones y paquetes en la importación. Por defecto es N.
- **DATAFILES:** Indica la lista de archivos de datos que se va a transferir a la base de datos.
- **TTS OWNERS:** Especifica el nombre o la lista de nombres de propietarios de datos en el espacio de tablas transportable.



## 13. Administración de Oracle II

### 13.3 Copias de seguridad



#### Caso práctico

- 5 Para este caso práctico es necesario ejecutar el script que se encuentra en el CD del libro y que crea los *tablespaces*: EJERCICIOS y TEMPEJER, los usuarios PRUEBA y USUEX\_IM con privilegios para hacer EXPORT e IMPORT, y las tablas NOTAS, ALUMNOS, PROFESORES, ASIGNATURAS y ASIG\_PROF, con sus relaciones y datos.

- C:\>EXP USERID=PRUEBA/PRUEBA, crea el archivo expdat.dmp en la carpeta desde donde se ejecuta el comando, con el esquema completo del usuario. Para importar el archivo será necesario tener un usuario con privilegio IMP\_FULL\_DATABASE, o DBA.

IMP USUEX\_IM/USUEX\_IM FROMUSER=PRUEBA FILE=C:\EXPDAT.DMP, importa todo lo de prueba al esquema de USUEX\_IM.

- C:\>EXP USERID=PRUEBA/PRUEBA FILE=C:\EXPORT\_IMPORT\3\_SECUN.DMP TABLES =(ASIGNATURAS) QUERY = \"WHERE CURSO = 3 AND ETAPA= 'SECUNDARIA' \", crea el archivo 3\_SECUN.DMP en la carpeta indicada, y contiene la tabla asignaturas con las asignaturas del curso tercero y de la etapa secundaria.

IMP USUEX\_IM/USUEX\_IM FROMUSER=PRUEBA FILE=C:\EXPORT\_IMPORT\3\_SECUN.DMP, importa la tabla ASIGNATURAS de tercero de secundaria del usuario PRUEBA al usuario USUEX\_IM.

- C:\>EXP SCOTT/TIGER FILE = C:\EXPORT\_IMPORT\DATOS\_SCOT.DMP OWNER=SCOTT GRANTS=N INDEXES=Y COMPRESS=Y ROWS=Y, SCOTT exporta todos sus objetos y crea el archivo DATOS\_SCOT.DMP, no se importan concesiones sobre objetos. Se utiliza compresión.

IMP USUEX\_IM/USUEX\_IM FROMUSER = SCOTT TOUSER=PRUEBA INDEXES=Y ROWS=Y FILE=C:\EXPORT\_IMPORT\DATOS\_SCOT.DMP. El usuario USUEX\_IM importa todo lo del usuario SCOTT al usuario PRUEBA, incluidos índices y filas.

C:\>IMP USUEX\_IM/USUEX\_IM FULL=Y NDEXFILE=C:\EXPORT\_IMPORT\INDEXSCOT.SQL FILE=C:\EXPORT\_IMPORT\DATOS\_SCOT.DMP. En este caso, se crea el archivo de índice INDEXSCOT.SQL, a partir del archivo de exportación. Este archivo se puede ejecutar aparte desde SQL haciendo algunos cambios en el mismo. Si se hace la importación sin índices pondremos: IMP USUEX\_IM/USUEX\_IM FROMUSER = SCOTT TOUSER=PRUEBA INDEXES=N ROWS=Y FILE=C:\EXPORT\_IMPORT\DATOS\_SCOT.DMP.

- C:\>EXP TRANSPORT\_TABLESPACE=Y TABLESPACES = (EJERCICIOS) FILE = C:\EXPORT\_IMPORT\TABL\_EJERCI.DMP LOG=C:\EXPORT\_IMPORT\LOGTAB\_EJER.LOG.

Exporta el *tablespace* EJERCICIOS. Este EXPORT debe hacerlo SYS o SYSTEM y conectarse como SYSDBA. Para poderlo transportar debe estar en modo READ ONLY. Es decir, antes el SYS debe ponerlo en sólo lectura:

```
SQL> CONNECT SYS/ARM AS SYSDBA
SQL> ALTER TABLESPACE EJERCICIOS READ ONLY;
```

Una vez realizada la exportación poner el *tablespace* en modo lectura escritura. SQL>ALTER TABLESPACE EJERCICIOS READ WRITE;

- C:\>EXP FULL=Y FILE=C:\BDCOMPLETA.DMP. Se hace una copia completa de la base de datos. Cuando pida usuario nos conectamos con SYS como SYSDBA:



### B. Copias de seguridad físicas

Éstas consisten en copiar los archivos que constituyen la base de datos. También se las denomina *copias de seguridad del sistema de archivos* ya que implican la utilización de comandos de copia de seguridad del sistema operativo. Oracle soporta dos métodos de copia de seguridad física: *fuerza de línea o en frío*, y *en línea o en caliente*.

Las **copias de seguridad fuera de línea o en frío** se producen cuando la base de datos se ha desconectado de modo normal y no debido a ningún fallo, es decir, la base de datos se ha cerrado con uno de estos comandos SHUTDOWN, SHUTDOWN IMMEDIATE o SHUTDOWN TRANSACTIONAL. Los archivos que deben copiarse son:

- Todos los archivos de datos.
- Todos los archivos de control.
- Todos los archivos de Redo Logs.

Y opcionalmente podemos copiar los archivos de parámetros de inicialización de la base de datos init.ora y spfile.ora. Hacer una copia de todos estos archivos proporciona una imagen exacta de la base de datos.

Así pues, para hacer una copia de una base de datos, primero hay que hacer SHUTDOWN y luego copiar en el destino elegido los archivos de datos que normalmente suelen ubicarse en: ORACLE\PRODUCT\10.1.0\ORADATA\ORCL.

Y los de parámetros en: ORACLE\PRODUCT\10.1.0\ADMIN\PFILE, los pfile, y en ORACLE\PRODUCT\10.1.0\DB\_1\DATABASE, los spfile.

Para saber los archivos de datos a copiar y su ubicación los consultamos en las siguientes vistas:

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

C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EJERCIC.DBF

SQL> select name from v$controlfile;
NAME

C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\CONTROL01.CTL
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\CONTROL02.CTL
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\CONTROL03.CTL
```



## 13. Administración de Oracle II

### 13.3 Copias de seguridad

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

C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO03.LOG
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO02.LOG
C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO01.LOG
```

Para restaurar esta base de datos copiamos la copia de seguridad realizada en los directorios correspondientes e iniciamos la base de datos con el comando STARTUP. La base de datos estará cerrada.

Las **copias de seguridad en línea o en caliente** se realizan con la base de datos abierta, para ello la base de datos debe estar en modo ARCHIVELOG para que los Redo Log se archiven creando registro de todas las transacciones realizadas. Recuerda que Oracle escribe en los Redo Log de manera cíclica: cuando se llena uno, pasa al siguiente, y así sucesivamente. Antes de sobrescribir cualquiera de ellos, el proceso ARCH realiza una copia de ese Redo Log. La mayoría de las bases de datos en producción deben ejecutarse en modo ARCHIVELOG. Mientras la base de datos está abierta puede realizarse una copia de:

- Todos los archivos de datos.
- Todos los archivos de Redo Logs archivados.
- Un archivo de control, mediante el comando **ALTER DATABASE**.

Para utilizar la función ARCHIVELOG, hay que poner la base de datos en modo ARCHIVELOG. Antes comprobamos el modo con `archive log list` (para todo esto hay que conectarse con SYS modo SYSDBA):

```
SQL> archive log list
Database log mode No Archive Mode
Automatic archival Disabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 226
Current log sequence 228
```

Por defecto, la base de datos está en modo NOARCHIVELOG, es decir, con la recuperación de medios deshabilitada. Para ponerla en ARCHIVELOG la base de datos debe estar cerrada y montada. Los pasos para pasar a modo ARCHIVELOG son:

- 1.<sup>º</sup> Se cierra la base de datos:

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

- 2.<sup>º</sup> Se monta la base de datos por defecto, la instancia ORCL. El *init* está en el directorio C:\ORACLE\PRODUCT\10.1.0\ADMIN\ORCL\PFILE:

```
SQL> startup mount;
```

## 13. Administración de Oracle II

### 13.3 Copias de seguridad



ORACLE instance started.

```
Total System Global Area 171966464 bytes
 Fixed Size 787988 bytes
 Variable Size 145750508 bytes
 Database Buffers 25165824 bytes
 Redo Buffers 262144 bytes
Database mounted.
```

Comprobamos que se monta la instancia ORCL:

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

| NAME  |
|-------|
| ----- |
| ORCL  |

3.<sup>º</sup> Activamos el modo ARCHIVELOG:

```
SQL> ALTER DATABASE ARCHIVELOG;
Database altered.
```

4.<sup>º</sup> A continuación, abrimos la base de datos y comprobamos si ha cambiado su estado:

```
SQL> ALTER DATABASE OPEN;
Database altered.
```

```
SQL> ARCHIVE LOG LIST;
```

| Database log mode            | Archive Mode              |
|------------------------------|---------------------------|
| Automatic archival           | Enabled                   |
| Archive destination          | USE_DB_RECOVERY_FILE_DEST |
| Oldest online log sequence   | 229                       |
| Next log sequence to archive | 231                       |
| Current log sequence         | 231                       |

La base de datos permanecerá en este modo hasta que se ponga en NOARCHIVELOG. La ubicación de los archivos de Redo Log archivados se determinan por el parámetro DB\_RECOVERY\_FILE\_DEST. Si editamos el *pfile* vemos que su ubicación es: C:\oracle\product\10.1.0\flash\_recovery\_area.

Para comprobar que se realizan copias de los archivos de Redo Log, hacemos cambios de log de forma manual:

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
SYSTEM ALTERED.
```

Luego comprobamos en el directorio C:\oracle\product\10.1.0\flash\_recovery\_area lo que se ha creado. Observa que se ha creado una carpeta con el nombre de la instancia, con una subcarpeta para ARCHIVELOG. En la Figura 13.1 se muestran varios Redo Logs archivados.



## 13. Administración de Oracle II

### 13.3 Copias de seguridad

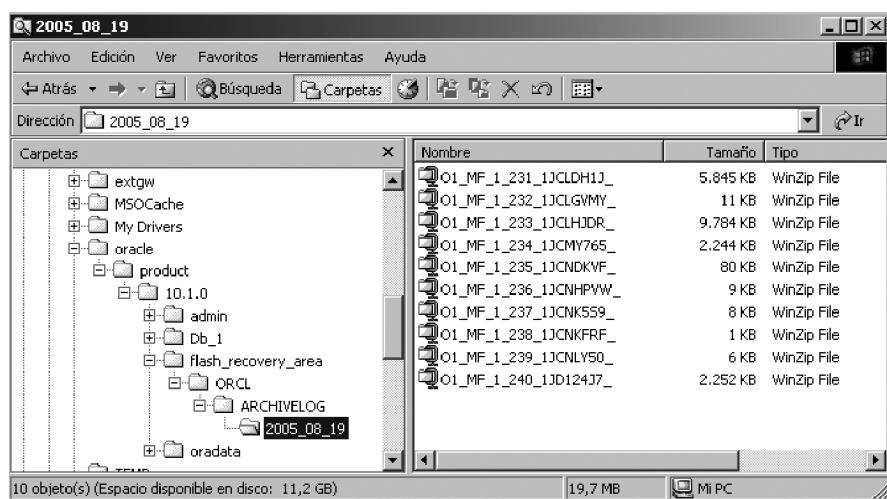


Figura 13.1. Ubicación de los Redo Logs archivados.

Para identificar a los archivos Oracle pone la extensión .ARC y un nombre, seguido del número de secuencia coincidente con Current log sequence. Se enumeran en el orden en que se crean.

Si hacemos un ARCHIVE LOG LIST, observamos que ha cambiado la secuencia y el número de secuencia actual, con respecto a la anterior ejecución. La vista V\$LOG indicará el log actual, y dirá si se han archivado a no:

| SQL> SELECT * FROM V\$LOG; |          |            |                 |          |           |                |                |                 |  |  |
|----------------------------|----------|------------|-----------------|----------|-----------|----------------|----------------|-----------------|--|--|
| GROUP#                     | THREAD#  | SEQUENCE#  | BYTES           | MEMBERS  | ARC       | STATUS         | FIRST_CHANGE#  | FIRST_TI        |  |  |
| 1                          | 1        | 233        | 10485760        | 1        | YES       | INACTIVE       | 5215596        | 19/08/05        |  |  |
| <b>2</b>                   | <b>1</b> | <b>234</b> | <b>10485760</b> | <b>1</b> | <b>NO</b> | <b>CURRENT</b> | <b>5215902</b> | <b>19/08/05</b> |  |  |
| 3                          | 1        | 232        | 10485760        | 1        | YES       | INACTIVE       | 5215561        | 19/08/05        |  |  |

|                              |                           |
|------------------------------|---------------------------|
| SQL> ARCHIVE LOG LIST;       |                           |
| Database log mode            | Archive Mode              |
| Automatic archival           | Enabled                   |
| Archive destination          | USE_DB_RECOVERY_FILE_DEST |
| Oldest online log sequence   | 232                       |
| Next log sequence to archive | 234                       |
| <b>Current log sequence</b>  | <b>234</b>                |

Una vez que la base de datos está en modo ARCHIVELOG y se ha provocado un cambio de log, se procederá a realizar las copias de seguridad de los *tablespaces*, de los archivos de Redo Log y del archivo de control.

- **Proceso para la copias de los tablespaces**

- Poner el *tablespace* en estado de copia ALTER TABLESPACE nombre BEGIN BACKUP. Esta orden hace que se congele la cabecera del archivo de datos y que se mantenga el número de secuencia, con lo que en caso de recuperación los logs se aplican a partir de este número.

## 13. Administración de Oracle II

### 13.3 Copias de seguridad



Aunque esté en modo BACKUP, el archivo de datos sigue estando disponible para el funcionamiento normal.

- Hacer copia de los archivos de datos asociados al *tablespace* (copiar y pegar).
- Dejar el *tablespace* en modo normal: ALTER TABLESPACE nombre END BACKUP.
- Forzar un punto de control para sincronizar las cabeceras de archivos mediante un cambio de log: ALTER SYSTEM SWITCH LOGFILE.

Todos estos pasos se repiten para cada *tablespace*.

### Caso práctico

#### 6 Realización de la copia de seguridad del *tablespace* USERS. Antes crea un directorio para los backups en C:\BACKUPS.

- Antes de hacer la copia vemos el fichero de datos asociado para luego copiarlo.

```
SQL> SELECT FILE_ID, FILE_NAME, TABLESPACE_NAME FROM DBA_DATA_FILES;
```

| FILE_ID | FILE_NAME                                           | TABLESPACE_NAME |
|---------|-----------------------------------------------------|-----------------|
| 4       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF   | USERS           |
| 3       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF  | SYSAUX          |
| 2       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF | UNDOTBS1        |
| 1       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF  | SYSTEM          |
| 5       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF | EXAMPLE         |
| 6       | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EJERCIC.DBF   | EJERCICIOS      |

- Iniciamos el backup con la orden:

```
SQL> ALTER TABLESPACE USERS BEGIN BACKUP;
Tablespace altered.
```

- Copiamos el fichero C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF en el directorio creado. La vista dinámica, con información de la copia de seguridad V\$BACKUP, nos devuelve los archivos que están en modo BACKUP, cuyo estado cambia a modo ACTIVE. La columna CHANGE# indica el número de *checkpoint*:

```
SQL> SELECT * FROM V$BACKUP;
```

| FILE# | STATUS     | CHANGE# | TIME     |
|-------|------------|---------|----------|
| 1     | NOT ACTIVE | 0       |          |
| 2     | NOT ACTIVE | 0       |          |
| 3     | NOT ACTIVE | 0       |          |
| 4     | ACTIVE     | 5218939 | 19/08/05 |
| 5     | NOT ACTIVE | 0       |          |
| 6     | NOT ACTIVE | 0       |          |

(Continúa)



## 13. Administración de Oracle II

### 13.3 Copias de seguridad

(Continuación)

- Terminada la copia, ponemos el *tablespace* a estado normal. La vista V\$BACKUP registrará los cambios.

```
SQL> ALTER TABLESPACE USERS END BACKUP;
Tablespace altered.
```

- **Proceso para las copias de los Redo Logs archivados**

Hacer copia de los Redo Log archivados ubicados en el directorio de archivelog. Después de la copia se pueden borrar.

- **Proceso para las copias del archivo de control**

Este archivo es uno de los que más debe protegerse. Su información es necesaria para el inicio de una instancia. La información de este archivo, como el archivo de Redo Log online actual o los nombres de los archivos de bases de datos, es necesaria para la recuperación de la instancia. Es conveniente mantener una copia de este archivo cada vez que hay un cambio en la configuración de los ficheros de datos. Los pasos para su copia son:

- 1.º Se crea una imagen del archivo con `ALTER DATABASE BACKUP CONTROLFILE TO <archivo_destino>`
- 2.º Se debe crear un archivo de comandos para reproducir el archivo de control con `ALTER DATABASE BACKUP CONTROLFILE TO TRACE`. Este archivo de traza permite recrear el archivo de control y se crea en el directorio indicado por el parámetro `USER_DUMP_DEST`.



### Caso práctico

#### 7 Realiza el backup del archivo de control.

- Primero creamos el archivo de traza, pero antes comprobamos la ubicación visualizando el parámetro `USER_DUMP_DEST`:

```
SQL> SHOW PARAMETER USER_DUMP_DEST
NAME TYPE VALUE

user_dump_dest string C:\ORACLE\PRODUCT\10.1.0\ADMIN\ORCL\UDUMP
```

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
Database altered.
```

Se crea el fichero de traza. Edita el archivo para ver su contenido. Su nombre suele llevar el nombre de la instancia, seguido de `ora` y un número, por ejemplo: `orcl_ora_876`.

- En segundo lugar, se hace la copia que guardaremos en el directorio `C:\BACKUPS`:

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO 'C:\BACKUPS\CCONTROL.BAK';
Database altered.
```



## 13.4 Restauración de datos

A la hora de recuperar los datos, ante fallos imprevistos, hay que tener en cuenta el modo en el que estaba la base de datos y las copias de que disponemos. El modo NOARCHIVELOG puede ser adecuado en bases de datos dedicadas a desarrollo, entrenamiento y prueba. En este caso, la restauración es sencilla. Basta con restaurar los archivos de la copia de seguridad en la ubicación de los archivos de la base de datos. El tiempo de recuperación es el tiempo que el hardware tarde en la restauración. El inconveniente de este modo es que los datos introducidos desde la última copia se pierden y sería necesario aplicarlos de nuevo.

### Caso práctico



**8** En este caso práctico, se va a simular un fallo en un archivo de datos: al abrir la base de datos ocurrirá un error. Para recuperarla procederemos a restaurar el archivo correspondiente. Antes de nada, comprueba que la base de datos está en NOARCHIVELOG; si no es así ponela en NOARCHIVELOG.

- En primer lugar, cerrar la base de datos con SHUTDOWN IMMEDIATE. Una vez cerrada hacemos copia de los archivos de datos (ya vimos cómo hacerlo en el apartado «Copias de seguridad en frío»).
- Seguidamente vamos a simular un fallo en el archivo USERS01.DBF. Por ejemplo, lo cambiamos de nombre o lo borramos.
- Arrancamos la base de datos con el comando STARTUP y ocurre lo siguiente:

```
SQL> startup
ORACLE instance started.

Total System Global Area 171966464 bytes
Fixed Size 787988 bytes
Variable Size 145750508 bytes
Database Buffers 25165824 bytes
Redo Buffers 262144 bytes
Database mounted.

ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF'
```

La base de datos llega a montarse, pero no se abre porque no encuentra un fichero de datos.

- Lo siguiente sería restaurar el archivo USERS01.DBF de los archivos de copia de seguridad. Basta con copiar y pegar.
- Cerrar la base de datos SHUTDOWN IMMEDIATE y volverla a abrir STARTUP. La base de datos abrirá sin ningún problema.

Nos imaginamos ahora que la restauración de este archivo se hace en una ubicación diferente porque el disco se ha estropeado. Lo hacemos en la carpeta D:\BASEORCL. En este caso, primero, se monta la instancia, luego utilizamos el comando ALTER DATABASE RENAME FILE para indicar la nueva ubicación de USERS01.DBF. De esta forma, el archivo de control se actualiza. Seguidamente se abre la base de datos:

```
SQL> STARTUP MOUNT
SQL> ALTER DATABASE RENAME FILE
```

(Continúa)



## 13. Administración de Oracle II

### 13.4 Restauración de datos

(Continuación)

```
'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF'
TO 'D:\BASEORCL\USERS01.DBF';
SQL> ALTER DATABASE OPEN;
```

Ojo: al hacer esto hemos cambiado la base de datos. El fichero USERS01.DBF está en su nueva ubicación. Si se desea colocarlo en la ubicación anterior hay que repetir estos pasos para ubicarlo en la carpeta habitual: parar la base de datos, hacer la copia en frío de D:\BASEORCL\USERS01.DBF y copiarlo en su carpeta habitual, montar la base de datos y hacer un ALTER DATABASE RENAME FILE 'D:\BASEORCL\USERS01.DBF' to 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF'. Si esto no se hace así ocurrirá un error de sincronización y Oracle pedirá recuperación de datos:

```
SQL> ALTER DATABASE OPEN;
ALTER DATABASE OPEN
*
ERROR at line 1:
ORA-01113: file 4 needs media recovery
ORA-01110: data file 4:
'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF'
SQL> recover datafile
'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF';
Media recovery complete.
SQL> ALTER DATABASE OPEN;
Database altered.
```

Oracle recuperará todas las transacciones que se encuentren en los archivos de redo (RED001.LOG, RED002.LOG, RED003.LOG), sin embargo, si estos archivos se sobreescreiben y la base de datos está en NOARCHIVELOG se perderán las transacciones no registradas en ellos.

### A. Recuperación en modo ARCHIVELOG

En el modo ARCHIVELOG podemos hacer recuperaciones completas de la base de datos y recuperaciones hasta un punto determinado.

En las recuperaciones completas es necesario tener todos los archivos de Redo Logs archivados desde que se realizó la copia de seguridad que se está utilizando en la restauración. Si falta alguno no se podrá realizar una restauración completa pues se aplican todos los archivos en secuencia. Este tipo de copia tiene las siguientes ventajas:

- Sólo es necesario restaurar los archivos perdidos o corruptos.
- Recupera todos los datos hasta el momento del fallo.
- El tiempo de recuperación es el que se tarda en restaurar los archivos perdidos y en aplicar todos los archivos de log archivados.
- **Recuperación completa ante fallos del medio físico**  
Antes de hacer recuperaciones de datos, en caso de errores, corrupción de discos y archivos de datos, es necesario disponer de una copia válida de los archivos de datos, los

## 13. Administración de Oracle II

### 13.4 Restauración de datos



logs desde que se realizó la copia de seguridad y los archivos de Redo Logs que aún no están archivados. Además, la base de datos debe estar definida en modo ARCHIVELOG.

Cumpliendo estos requisitos, el proceso para recuperar los archivos de datos es el siguiente:

- Comprueba que los archivos que se van a sobrescribir no están abiertos durante la recuperación. Consulta las vistas: V\$DATAFILE y V\$TABLESPACE para comprobar el estado del archivo.
- Sólo se debe restaurar el archivo perdido o dañado.
- Los archivos de Redo Log online no se restauran.
- La base de datos se pondrá en modo MOUNT o en modo OPEN.
- Utilizaremos el comando RECOVER para recuperar los datos:
  - Con la base de datos montada podremos utilizar: RECOVER DATABASE, RECOVER DATAFILE o ALTER DATABASE RECOVER DATABASE.
  - Con la base de datos abierta podremos utilizar: RECOVER TABLESPACE, RECOVER DATAFILE o ALTER DATABASE RECOVER DATAFILE.

Para realizar una recuperación completa utilizaremos los siguientes métodos:

- **Recuperación con la base de datos abierta.** En este caso, se pierde un archivo de datos al que no pertenecen ni el sistema ni el *tablespace* de Rollback. Los ficheros de datos se pondrán OFFLINE, arrancará la base de datos, se realizará la recuperación de ese archivo, y seguidamente el archivo se pondrá ONLINE.

#### Caso práctico

- 9 **Recuperación de un *datafile* con la base de datos abierta.** Antes de realizar el caso práctico vamos a comprobar el modo de apertura de la base de datos con ARCHIVE LOG LIST o consultando V\$DATABASE. También hay que tener un backup de todos los *tablespaces*, como se hizo en el apartado de copias en caliente, con la base de datos arrancada.

- En primer lugar, se hacen copias de los archivos de datos y los *tablespaces* con el usuario SYS, para ello consultamos la vista: SQL> SELECT FILE\_ID, FILE\_NAME, TABLESPACE\_NAME FROM DBA\_DATA\_FILES.
- Iniciamos el backup de los *tablespaces*:

```
SQL> ALTER TABLESPACE USERS BEGIN BACKUP;
SQL> ALTER TABLESPACE SYSAUX BEGIN BACKUP;
SQL> ALTER TABLESPACE UNDOTBS1 BEGIN BACKUP;
SQL> ALTER TABLESPACE SYSTEM BEGIN BACKUP;
SQL> ALTER TABLESPACE EXAMPLE BEGIN BACKUP;
```

- Copiamos los ficheros asociados a los *tablespaces* en el directorio de copias C :\BACKUPS.

(Continúa)



## 13. Administración de Oracle II

### 13.4 Restauración de datos

(Continuación)

- Una vez copiados, finalizamos la copia poniendo:

```
SQL> ALTER TABLESPACE USERS END BACKUP;
SQL> ALTER TABLESPACE SYSAUX END BACKUP;
SQL> ALTER TABLESPACE UNDOTBS1 END BACKUP;
SQL> ALTER TABLESPACE SYSTEM END BACKUP;
SQL> ALTER TABLESPACE EXAMPLE END BACKUP;
```

- Creamos la tabla NUEEMPLE en el esquema de Scott, a partir de la tabla EMP. Vamos a ir añadiendo registros y provocando cambios de log. Iremos apuntando los números de secuencia de los mismos para cuando se desea recuperar a partir de un log.

```
CREATE TABLE SCOTT.NUEEMPLE AS SELECT * FROM SCOTT.EMP;
```

Comprobamos el *tablespace* donde se ha creado esta tabla para asegurarnos que es USERS y para saber el fichero de datos a recuperar:

```
SQL> SELECT TABLESPACE_NAME FROM DBA_TABLES WHERE TABLE_NAME = 'NUEEMPLE' AND OWNER = 'SCOTT';

SQL> INSERT INTO SCOTT.NUEEMPLE SELECT * FROM SCOTT.EMP;
SQL> COMMIT;
SQL> SELECT COUNT(*) FROM SCOTT.NUEEMPLE;
COUNT(*)

28
SQL> SELECT * FROM V$LOG;
GROUP# THREAD# SEQUENCE# BYTES MEMBERS ARC STATUS FIRST_CHANGE# FIRST_TI
----- ----- ----- ----- ----- ----- -----
1 1 251 10485760 1 NO ACTIVE 5331863 22/08/05
2 1 249 10485760 1 YES INACTIVE 5302660 22/08/05
3 1 250 10485760 1 NO CURRENT 5324174 22/08/05
```

En la secuencia de log 250, el CURRENT, SCOTT.NUEEMPLE tiene 28 registros. Hacemos COMMIT, cambiamos de log y los apuntamos para luego recuperar la tabla en un momento determinado.

```
SQL> COMMIT;
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> INSERT INTO SCOTT.NUEEMPLE SELECT * FROM SCOTT.EMP;
SQL> SELECT COUNT(*) FROM SCOTT.NUEEMPLE;
SQL> SELECT * FROM V$LOG;
SQL> COMMIT;
```

En la secuencia 251 la tabla SCOTT.NUEEMPLE tiene 42 registros.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> INSERT INTO SCOTT.NUEEMPLE SELECT * FROM SCOTT.EMP;
SQL> SELECT COUNT(*) FROM SCOTT.NUEEMPLE;
SQL> SELECT * FROM V$LOG;
SQL> COMMIT;
```

(Continúa)

## 13. Administración de Oracle II

### 13.4 Restauración de datos



(Continuación)

En la secuencia 252 la tabla SCOTT.NUEMPLE tiene 56 registros. Si se abre la carpeta C:\ORACLE\PRODUCT\10.1.0\FLASH\_RECOVERY\_AREA\ORCL\ARCHIVELOG\fecha\_archivado\ se podrán ver los logs creados. En la Figura 13.2 se ven los logs del ejercicio. La fecha es la del 22 de agosto de 2005:

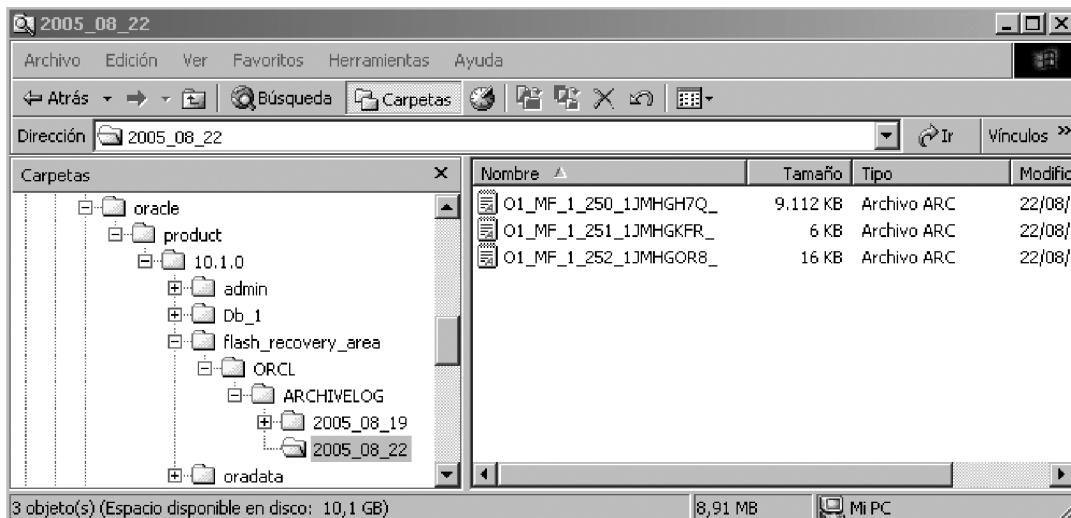


Figura 13.2. Logs archivados del ejercicio.

- A continuación, cerramos la base de datos con SHUTDOWN IMMEDIATE y provocamos un error en la base de datos: borramos el archivo USERS01.DBF, arrancamos y vemos que la base de datos se monta pero no arranca pues no localiza el archivo. Lo que hay que hacer es poner el *datafile* fuera de línea OFFLINE, restaurar la copia de seguridad, abrir la base de datos y recuperar el archivo:

```
SQL> ALTER DATABASE DATAFILE
'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF' OFFLINE;
```

O también **ALTER DATABASE DATAFILE 4 OFFLINE;** (pues 4 es el número de este archivo). Al poner el archivo OFFLINE, permite que los usuarios que no trabajen con este archivo puedan conectarse y realizar transacciones.

Restauramos la copia de seguridad y abrimos la base de datos.

```
SQL> ALTER DATABASE OPEN;
Ponemos el archivo a ONLINE, pero ocurre un error y Oracle indica que el archivo se tiene que recuperar:
```

```
SQL> ALTER DATABASE DATAFILE 4 ONLINE;
ALTER DATABASE DATAFILE 4 ONLINE
*
ERROR at line 1:
ORA-01113: file 4 needs media recovery
ORA-01110: data file 4: 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF'
```

Se restaura con la orden RECOVER DATAFILE, para ver los archivos de datos que necesitan recuperación consultamos la vista: SELECT \* FROM V\$RECOVER\_FILE;

(Continúa)



## 13. Administración de Oracle II

### 13.4 Restauración de datos

(Continuación)

```
SQL> RECOVER DATAFILE 4
ORA-00279: change 5331190 generated at 08/22/2005 13:42:39 needed for thread 1
ORA-00289: suggestion :
C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_22\01_MF_1_2
50_%U_.ARC
ORA-00280: change 5331190 for thread 1 is in sequence #250
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
Media recovery complete.
```

Aparece un mensaje en el que Oracle nos pregunta a partir de qué log va a aplicar para iniciar la recuperación. En el ejemplo, Oracle sugiere que empecemos aplicar el log 250 (es el primer log que se creó y archivó anteriormente), pulsamos *Enter* para aceptarlo, y esperamos a que se realice la restauración completa.

- A continuación, se pone **ONLINE** el *datafile* y comprobamos el contenido de la tabla NUEEMPLE de Scott, debe tener 56 registros.

```
SQL> ALTER DATABASE DATAFILE 4 ONLINE;
SQL> SELECT COUNT(*) FROM SCOTT.NUEEMPLE;
```

La tabla NUEEMPLE contiene todos los registros, ya que hemos aplicado todos los log.

- Podemos recuperar a partir de un log determinado. Sin embargo, al estar la base en modo ARCHIVELOG va a pedir toda la secuencia de log. En el ejemplo que tenemos, si deseamos recuperar a partir del log 251, nos dará error y nos indicará que necesita el 250 para recuperar el 251. En **Specify log** indicamos la ruta y el nombre de log:

```
SQL> recover datafile 4
ORA-00279: change 5331190 generated at 08/22/2005 13:42:39 needed for thread 1
ORA-00289: suggestion :
C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_22\01_MF_1_2
50_%U_.ARC
ORA-00280: change 5331190 for thread 1 is in sequence #250
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_22\01_MF_1_2
51_1JMHGKFR_.ARC
ORA-00310: archived log contains sequence 251; sequence 250 required
ORA-00334: archived log:
'C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_22\01_MF_1_
251_1JMHGKFR_.ARC'
```

Recuperar el archivo y ponerlo **ONLINE**.

- **Recuperación de una base de datos cerrada.** Este método utiliza los comandos RECOVER DATABASE o RECOVER DATAFILE cuando la base de datos puede dejar de funcionar durante un tiempo, o cuando los archivos recuperados pertenecen al sistema o al *tablespace* del segmento de Rollback, o cuando es necesario recuperar toda la base de datos o la mayoría de los archivos de datos.

## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN



## 13.5 Copias de seguridad y recuperación con RMAN

**RMAN** es una utilidad que permite hacer rápidamente backup de todos los datos o de una parte de la base de datos. Es un potente lenguaje de comandos independiente del sistema operativo. Utiliza los procesos de Oracle para realizar las copias y las recuperaciones, de ahí que se denominen *operaciones de copia de seguridad y recuperación gestionadas por el servidor*. El RMAN tiene 4 componentes:

- **El ejecutable RMAN**, que se encuentra en el directorio bin de la carpeta de Oracle, y que se ejecuta desde el sistema operativo simplemente tecleando `rman`. Se pueden ver los parámetros y argumentos tecleando RMAN HELP. Ver Figura 13.3.

The screenshot shows a Windows Command Prompt window titled "Seleccionar C:\WINNT\system32\cmd.exe". The command entered is "C:\>rman help". The output displays a table of arguments and their descriptions, followed by error messages and a copyright notice.

| Argument  | Value         | Description                                |
|-----------|---------------|--------------------------------------------|
| target    | quoted-string | connect-string for target database         |
| catalog   | quoted-string | connect-string for recovery catalog        |
| nocatalog | none          | if specified, then no recovery catalog     |
| cmdfile   | quoted-string | name of input command file                 |
| log       | quoted-string | name of output message log file            |
| trace     | quoted-string | name of output debugging message log file  |
| append    | none          | if specified, log is opened in append mode |
| debug     | optional-args | activate debugging                         |
| msgno     | none          | show RMAN-nnnn prefix for all messages     |
| send      | quoted-string | send a command to the media manager        |
| pipe      | string        | building block for pipe names              |
| timeout   | integer       | number of seconds to wait for pipe input   |

Both single and double quotes <' or "> are accepted for a quoted-string.  
Quotes are not required unless the string contains embedded white-space.

RMAN-00571: =====  
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====  
RMAN-00571: =====  
RMAN-00552: syntax error in command line arguments  
RMAN-01005: syntax error: found "identifiem": expecting one of: "append, at, aux  
iliary, catalog, cmdfile, clone, debug, log, msglog, mask, msgno, nocatalog, pip  
e, recat, script, slaxdebug, send, target, timeout, trace"  
RMAN-01008: the bad identifier was: help  
RMAN-01007: at line 2 column 1 file: command line arguments

C:\>

Figura 13.3. Argumentos de RMAN.

- **Una o más bases de datos de destino**, que son las bases de datos para las que se realizan operaciones de copia y restauración. La información del archivo de control de la base de datos de destino es utilizada por los procesos de servidor que llama RMAN en las operaciones de copia y restauración. La base de datos se especifica con el argumento TARGET. Pondremos RMAN TARGET ORCL para conectarnos a la instancia ORCL. El usuario que se conecta debe tener privilegios SYSDBA, si no ponemos usuario presupone SYS:

```
C:\>RMAN TARGET ORCL
Recovery Manager: Release 10.1.0.2.0 - Production
Copyright (c) 1995, 2004, Oracle. All rights reserved.
target database Password:
connected to target database: ORCL (DBID=1086216171)
```

O también: C:\>RMAN TARGET SYS/ARM@ORCL



## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN

- **El catálogo de recuperación de RMAN** también llamado **Metadatos de RMAN**. Los datos que RMAN utiliza para las operaciones de copia, recuperación y restauración, conocidos como los metadatos de RMAN, se almacenan en el archivo de control de la base de datos destino. Por esta razón, no es necesario crear un catálogo de recuperación. Sin embargo, resulta útil configurarlo pues muchas de las funciones, como los archivos de comandos almacenados y la copia de seguridad automática, no están disponibles sin el catálogo de recuperación.

Se recomienda ubicarlo en una base de datos distinta a la de destino; en este caso, se crea en la misma base de datos. El catálogo lo debe crear un usuario con el rol RECOVERY\_CATALOG\_OWNER, el administrador creará el usuario RMAN, por ejemplo:

```
SQL> CREATE USER RMAN IDENTIFIED BY RMAN
TEMPORARY TABLESPACE TEMP
DEFAULT TABLESPACE USERS;
SQL> GRANT RECOVERY_CATALOG_OWNER, CONNECT, RESOURCE
TO RMAN;
SQL> ALTER USER RMAN QUOTA UNLIMITED ON USERS;
```

Para crear el catálogo entramos como usuario RMAN a la instancia ORCL. Se crea un archivo de log CATALOG.LOG para ver si la creación se ha realizado correctamente, y escribimos lo siguiente:

```
C:\>RMAN CATALOG RMAN/RMAN@ORCL MSGLOG=C:\CATALOG.LOG
RMAN> CREATE CATALOG
RMAN> EXIT
```

Y para conectarse al catálogo pondremos:

```
C:\>RMAN TARGET ORCL
Recovery Manager: Release 10.1.0.2.0 - Production
Copyright (c) 1995, 2004, Oracle. All rights reserved.
target database Password:
connected to target database: ORCL (DBID=1086216171)
RMAN> CONNECT CATALOG RMAN/RMAN@ORCL
connected to recovery catalog database
```

Lo siguiente que se hace es registrar la base de datos destino en el catálogo, para almacenar la información sobre la base de datos. Si la base de datos no se registra en el catálogo, éste no se podrá utilizar para almacenar información sobre la base de datos:

```
RMAN> REGISTER DATABASE;
database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

- **Software de gestión de soportes físicos.** Lo utiliza RMAN para escribir y leer en soportes físicos como cintas. Este software es proporcionado por los fabricantes de sistemas de almacenamiento y soportes físicos.

## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN



Existen una serie de comandos que nos van a permitir analizar la información del catálogo de recuperación:

- REPORT SCHEMA, visualiza la estructura de la base de datos.
- REPORT NEED BACKUP, indica de qué archivos es necesario hacer copias de seguridad:

```
RMAN> REPORT NEED BACKUP;
RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Report of files with less than 1 redundant backups
```

| File | #bkps | Name                                                |
|------|-------|-----------------------------------------------------|
| 1    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF  |
| 2    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF |
| 3    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF  |
| 4    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF   |
| 5    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF |
| 6    | 0     | C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EJERCIC.DBF   |

- REPORT OBSOLETE, indica qué copias de seguridad se pueden eliminar.

### A. El comando BACKUP

Se puede ejecutar directamente o entre las llaves de un comando RUN. La salida puede escribirse en cinta o disco. Este comando crea juegos de copias de seguridad que normalmente contienen más de un archivo. Las copias de seguridad no contienen bloques de datos vacíos. Para extraer archivos de la copia de seguridad se realizará una operación de restauración. Los juegos de copias de seguridad de datos pueden ser incrementales o completos.

El comando puede realizar una copia de seguridad de una base de datos, un *tablespace*, un archivo de datos o de los Redo Logs archivados:

- BACKUP DATABASE; copia la base de datos.
- BACKUP TABLESPACE *nombre*; copia el *tablespace* indicado, por ejemplo:  
BACKUP TABLESPACE USERS;
- BACKUP ARCHIVELOG ALL; realiza copia de seguridad de los Redo Logs archivados.
- BACKUP DATAFILE *nombre*; realiza una copia de un archivo de datos, por ejemplo: BACKUP DATAFILE "C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF";

La orden LIST BACKUP; muestra información detallada sobre los archivos de copias de seguridad disponibles.



## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN



#### Caso práctico

##### 10 Deseamos hacer un backup completo de la base de datos.

- Nos conectamos a RMAN con SYS utilizando el catálogo creado:

```
C:\> RMAN TARGET SYS/ARM@ORCL CATALOG RMAN/RMAN@ORCL
```

- Ejecutamos la orden BACKUP DATABASE, observa que copia los archivos de datos, el control file y el spfile. Observa también que lo ubica en el directorio FLASH\_RECOVERY\_AREA, es la ubicación predeterminada y a cada archivo de copia creado le asigna un nombre:

```
RMAN> BACKUP DATABASE;
Starting backup at 23/08/05
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=143 devtype=DISK
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF
input datafile fno=00003 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF
input datafile fno=00005 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF
input datafile fno=00002 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF
input datafile fno=00006 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EJERCIC.DBF
input datafile fno=00004 name=C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 23/08/05
channel ORA_DISK_1: finished piece 1 at 23/08/05
piece handle=C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\BACKUPSET\2005_08
_23\01_MF_NNNDF_TAG20050823T131605_1JP1CR70_.BKP comment=None
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:47
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current controlfile in backupset
including current SPFILE in backupset
channel ORA_DISK_1: starting piece 1 at 23/08/05
channel ORA_DISK_1: finished piece 1 at 23/08/05 piece
handle=C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\BACKUPSET\2005_08
_23\01_MF_NCSNF_TAG20050823T131605_1JP1H3HX_.BKP comment=None
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:06
Finished backup at 23/08/05
```

Una vez que tenemos el backup vamos a provocar un fallo y restaurar la base de datos. Utilizaremos los comandos RESTORE y RECOVER. Cerramos la base de datos y borramos dos ficheros de datos, por ejemplo, USERS01.DBF y SYSTEM01.DBF, ojo: borramos el SYSTEM.

- Desde SQL\*PLUS, con el usuario SYS as SYSDBA, cerramos la base de datos SHUTDOWN IMMEDIATE. También se puede cerrar la base de datos y abrirla desde RMAN, si hemos entrado con un usuario SYSDBA.
- Borramos los archivos de datos indicados.

(Continúa)

## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN



(Continuación)

- Si intentamos conectarnos a la instancia y al catálogo nos va a dar error pues la base de datos no funciona (por eso se recomienda crear el catálogo en otra base de datos). Así pues, nos conectamos a RMAM con el usuario SYS y sin catálogo (la información de la base de datos la obtendrá del archivo de control): RMAN TARGET SYS/ARM. Montamos la base de datos, restauramos con RESTORE y recuperamos la sincronización de los archivos con RECOVER:

```
C:\>RMAN TARGET SYS/ARM
Recovery Manager: Release 10.1.0.2.0 - Production
Copyright (c) 1995, 2004, Oracle. All rights reserved.
connected to target database (not started)

RMAN> STARTUP MOUNT;

Oracle instance started
database mounted
Total System Global Area 171966464 bytes
Fixed Size 787988 bytes
Variable Size 145750508 bytes
Database Buffers 25165824 bytes
Redo Buffers 262144 bytes

RMAN> RESTORE DATABASE;
Starting restore at 23/08/05
using target database controlfile instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=160 devtype=DISK
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00002 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF
restoring datafile 00003 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF
restoring datafile 00004 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF
restoring datafile 00005 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF
restoring datafile 00006 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EJERCIC.DBF
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\BACKUPSET\2005_08
_23\01_MF_NNNDF_TAG20050823T131605_1JP1CR70_.BKP tag=TAG20050823T131605
channel ORA_DISK_1: restore complete
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\BACKUPSET\2005_08
_23\01_MF_NNNDF_TAG20050823T134715_1JP363WG_.BKP tag=TAG20050823T134715
channel ORA_DISK_1: restore complete
Finished restore at 23/08/05

RMAN> RECOVER DATABASE;
Starting recover at 23/08/05
using channel ORA_DISK_1
starting media recovery
archive log thread 1 sequence 260 is already on disk as file C:\ORACLE\PRODUCT\1
0.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_23\01_MF_1_260_1JP3Y580_.ARC
```

(Continúa)



## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN

(Continuación)

```
archive log thread 1 sequence 261 is already on disk as file C:\ORACLE\PRODUCT\1
0.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_23\01_MF_1_261_1JP9Q3QV_.ARC
archive log thread 1 sequence 262 is already on disk as file C:\ORACLE\PRODUCT\1
0.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2005_08_23\01_MF_1_262_1JPB3VXM_.ARC
archive log filename=C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\ORCL\ARCHIVELO
G\2005_08_23\01_MF_1_260_1JP3Y580_.ARC thread=1 sequence=260
media recovery complete
Finished recover at 23/08/05
```

- Finalmente, la base de datos se abre llamando a SQL:

```
RMAN> SQL "ALTER DATABASE OPEN";
sql statement: alter database open
```

### B. Tipos de copias de seguridad

A la hora de hacer copias de seguridad se puede elegir el tipo de copia a realizar. Existen varios tipos:

- **Las copias de seguridad completas.** Contienen todos los bloques de archivos de datos. Ademas pueden contener copias imagen, logs archivados o archivos de control.
- **Las copias de seguridad incrementales** (Nivel  $\geq 0$ ). Contienen sólo los bloques modificados del mismo nivel, es decir, los bloques que han sido modificados desde la copia de seguridad incremental anterior, de nivel menor o igual. Una copia incremental de nivel  $n$  copia todos los bloques modificados desde la copia de nivel anterior o igual. La primera copia que debe crearse es de nivel 0, después se realizarán las incrementales basadas en los cambios de la copia de nivel 0. Las copias deben crearse en orden de niveles: 0, 1, 2 y 3. Oracle utiliza 4 niveles de copia incremental:
  - Nivel 0, es la copia inicial de base.
  - Nivel 1 para copias incrementales mensuales, que copia los datos cambiados desde el último mes nivel 1, o desde la última incremental base de nivel 0.
  - Nivel 2, copia semanal que incluye los bloques cambiados desde la última copia semanal nivel 2, o de nivel 1.
  - Nivel 3, es el nivel diario. Copia los bloques cambiados desde la última copia diaria nivel 3, o de nivel 2.

Las copias de seguridad incrementales de varios niveles facilitan las operaciones de recuperación puesto que durante la restauración sólo se necesita una copia de seguridad incremental de un nivel específico.

## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN



#### Caso práctico

- 11 Estamos manteniendo una base de datos en una máquina que tiene pocos recursos y que, en hacer una copia, se tarda 4 horas. Nuestra base de datos trabaja durante las 24 horas del día, todos los días de la semana, por esa razón no se pueden realizar copias de nivel 0 (que guardan modificaciones desde el último 0) más de una vez a la semana. Se necesita que la recuperación sea rápida en caso de fallo. Así pues, se decide la siguiente estrategia de copia para la semana:

- Realizar una copia de nivel 0 el día de la semana de menor actividad, que puede ser el domingo. A partir de esta, se van creando las incrementales.

**BACKUP INCREMENTAL LEVEL = 0 (DATABASE);**

- Todos los días de la semana, excepto el miércoles, se realizará una copia de nivel 2. Para guardar los cambios diarios:

**BACKUP INCREMENTAL LEVEL = 2 (DATABASE);**

- Se decide hacer una copia de nivel 1 el miércoles, que copiará los bloques cambiados desde el domingo, es decir, los niveles iguales o inferior, pues es un día en el que la actividad es floja. De esta forma, si hay un fallo el viernes, sólo es necesario restaurar el domingo, miércoles y jueves (los cambios del lunes y el martes van incluidos en los del miércoles).

**BACKUP INCREMENTAL LEVEL = 1 (DATABASE);**

Las copias de nivel 0 deben conservarse hasta que se realice otra del mismo nivel. La recuperación de copias las podemos representar como muestra la Figura 13.4.

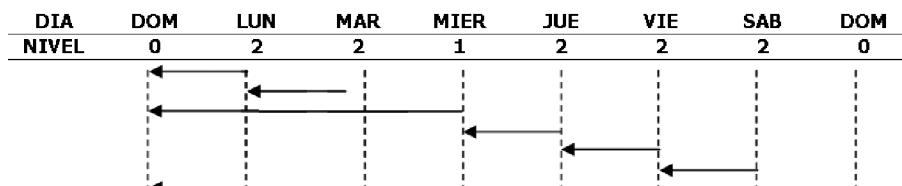


Figura 13.4. Representación para la recuperación de copias incrementales.

Las copias de seguridad incrementales sólo son aplicables si la base de datos de destino está abierta en modo ARCHIVELOG. También se pueden hacer copias incrementales de tablespaces y ficheros de datos.

- Las copias de seguridad acumulativas incrementales (Nivel  $\geq 0$ ) copian todos los bloques modificados desde la copia de seguridad incremental anterior de nivel menor a  $n$ . Pueden tardar más que las otras copias pues escriben más bloques y generan más archivos de copia y de mayor tamaño. Este tipo de copias aumenta la velocidad de recuperación, pues el número de copias a aplicar es menor. Por ejemplo, si se realiza una copia de seguridad incremental de nivel 2, la siguiente acumulativa copiará todos los bloques modificados de nuevo más los copiados mediante la copia de seguridad incremental de nivel 2. Esto significa que sólo será necesaria una copia de seguridad incremental del mismo nivel para una recuperación completa. El comando para crear copias acumulativas es:

**BACKUP INCREMENTAL LEVEL = 2 CUMULATIVE (DATABASE);**



## 13. Administración de Oracle II

### 13.5 Copias de seguridad y recuperación con RMAN

En la Figura 13.5 se muestra un esquema de las copias a realizar para recuperar una base de datos. Los martes, jueves, viernes y sábados las copias son acumulativas. Domingo, lunes y miércoles incrementales.

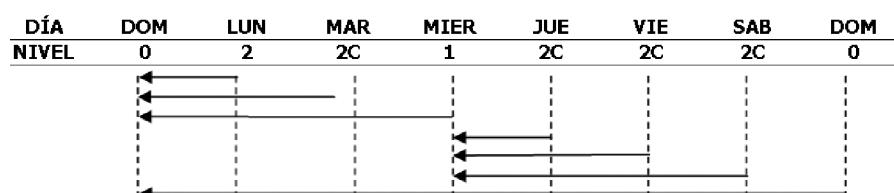


Figura 13.5. Recuperación de copias incrementales acumulativas.

### C. Restauración y recuperación

A la hora de restaurar y recuperar hay que tener en cuenta las copias disponibles y su ubicación. Se podrán realizar restauraciones de la base de datos (vista anteriormente en el caso práctico), de tablespaces y de archivos de datos.

Si las copias de seguridad han sido realizadas con RMAN sólo pueden ser restauradas con RMAN. Para restaurar una base de datos en estado MOUNT utilizaremos:

- **RESTORE:** este comando restaura archivos de copias de seguridad en la ubicación predeterminada o en otra ubicación si se indica.

RESTORE DATABASE; restaura la base de datos completa.

RESTORE TABLESPACE *nombre*; restaura un *tablespace*, antes de restaurarlo hay que ponerlo offline, por ejemplo:

```
RMAN> SQL "ALTER TABLESPACE USERS OFFLINE IMMEDIATE";
RMAN> RESTORE TABLESPACE USERS;
```

- **RECOVER:** aplica registros de Redo Logs o copias de seguridad incrementales si las hubiera a un conjunto de copias restaurado o a una copia normal, con el fin de actualizar la base de datos a un instante específico y sincronizar todos los archivos de datos.

RECOVER DATABASE; recupera y sincroniza la base de datos completa.

RECOVER TABLESPACE *nombre*; después de hacer recover hay que ponerlo online:

```
RMAN> SQL "ALTER TABLESPACE USERS OFFLINE IMMEDIATE";
RMAN> RESTORE TABLESPACE USERS;
RMAN> RECOVER TABLESPACE USERS;
RMAN> SQL "ALTER TABLESPACE USERS ONLINE";
```

En este apartado se ha visto una pequeña parte de lo que se puede hacer con RMAN. Esta utilidad tiene una gran cantidad de comandos para realizar las copias de seguridad, restaurar y recuperar una base de datos.



## 13.6 Análisis de los Redo Logs

Oracle dispone de una utilidad que va a permitir analizar los archivos de Redo Logs, con lo que se podrá hacer un seguimiento de todos los cambios que se producen en la base de datos y en el diccionario de datos. La utilidad se llama **LOGMINER** y ofrece la posibilidad de procesar los archivos de Redo Log y traducir su contenido en sentencias SQL que representan las operaciones lógicas realizadas en la base de datos.

LOGMINER requiere un diccionario de datos para traducir los contenidos de los Redo Logs. El usuario SYS crea el diccionario. Para crear el diccionario de datos hacemos lo siguiente:

- Editar el archivo de inicialización de la base de datos el **SPFILEORCL.ORA** (para la instancia ORCL) que se encuentra en el directorio C:\ORACLE\PRODUCT\10.1.0\DB\_1\DATABASE.

Añadir la inicialización del parámetro **UTL\_FILE\_DIR**, donde indicaremos un directorio para ubicar el archivo del diccionario de datos, es decir, un directorio donde LOGMINER guardará las salidas de datos que genere. Crear un directorio llamado LOGMINER dentro de la instancia ORCL, y ese será el UTL\_FILE\_DIR: \*.utl\_file\_dir='C:\oracle\product\10.1.0\oradata\orcl\logminer'. Antes de cambiar el SPFILE hacer una copia por si hubiera problemas. Observa en la Figura 13.6 cómo modificar el SPFILEORCL.

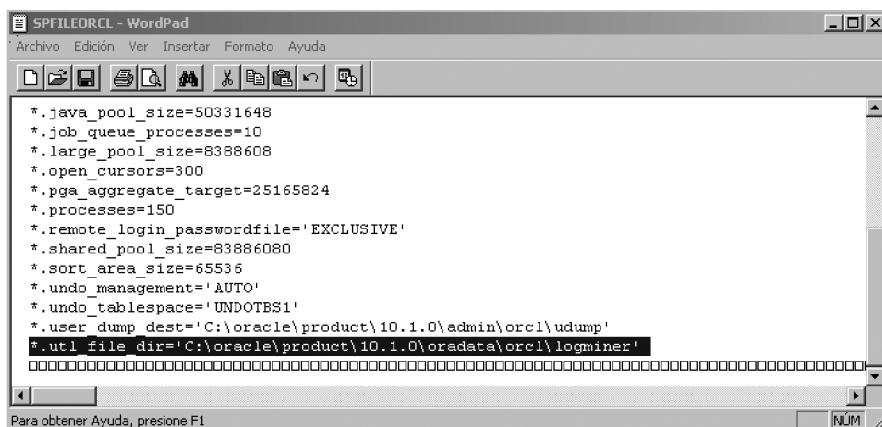


Figura 13.6. Modificación del archivo SPFILEORCL.ORA.

También se puede cambiar con el comando ALTER SYSTEM. En ambos casos hay que parar e iniciar la base de datos. Conectado como SYS para cambiar el parámetro, primero comprobamos el valor, ejecutamos el comando ALTER SYSTEM, paramos y arrancamos la base de datos y comprobamos si el parámetro se ha actualizado:

```

SQL> show parameter utl_
NAME TYPE VALUE

create_stored_outlines string
utl_file_dir string

```



## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs

```
SQL> alter system set utl_file_dir =
'C:\oracle\product\10.1.0\oradata\orcl\logminer' SCOPE=spfile;

System altered.
SQL> show parameter utl_
NAME TYPE VALUE

create_stored_outlines string
utl_file_dir string

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
Total System Global Area 171966464 bytes
Fixed Size 787988 bytes
Variable Size 145750508 bytes
Database Buffers 25165824 bytes
Redo Buffers 262144 bytes
Database mounted.
Database opened.
SQL> show parameter utl_
NAME TYPE VALUE

create_stored_outlines string
utl_file_dir string C:\oracle\pro-
 duct\10.1.0\
 oradata\orcl\
 logminer
```

- Ejecutar el procedimiento DBMS\_LOGMNR\_D.BUILD, que se encargará de crear el archivo de diccionario que vamos a llamar ORCLDICT.ORA, y lo va a guardar en la carpeta indicada. La opción STORE\_IN\_FLAT\_FILE significa que el archivo no va a tener formato. Una vez creado el diccionario se podrán analizar los Redo Logs. El procedimiento a ejecutar es:

```
SQL> EXECUTE
DBMS_LOGMNR_D.BUILD('ORCLDICT.ORA', 'C:\ORACLE\PRO-
DUCT\10.1.0\ORADATA\ORCL\LOGMINER',OPTIONS=>DBMS_LOGMNR_
D.STORE_IN_FLAT_FILE);
```

#### A. Sesión de LogMiner

Una vez creado el diccionario y antes de empezar a analizar los Redo Logs hay que configurar una sesión de LogMiner. Debemos especificar los archivos de log que se desean analizar. Esta utilidad puede analizar los archivos de log archivados y los online. Para ello se utilizará el procedimiento DBMS\_LOGMNR.ADD\_LOGFILE, que añade un archivo a

## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs



la lista. Así pues, nos conectamos con SYS y primero se crea una lista nueva con el primer log a analizar:

```
SQL> EXECUTE
DBMS_LOGMNR.ADD_LOGFILE ('C:\ORACLE\PRODUCT\10.1.0\ORA-
DATA\ORCL\O1_MF_1_250.arc', DBMS_LOGMNR.NEW);
```

Si se desea añadir otro log a la lista pondremos:

```
SQL> EXECUTE
DBMS_LOGMNR.ADD_LOGFILE ('C:\ORACLE\PRODUCT\10.1.0\ORA-
DATA\ORCL\REDO02.LOG', DBMS_LOGMNR.ADDFILE);
```

Y si se desea borrar un archivo de la lista utilizaremos la opción: **DBMS\_LOGMNR.REMOVEFILE**

```
SQL> EXECUTE
DBMS_LOGMNR.ADD_LOGFILE ('C:\ORACLE\PRODUCT\10.1.0\ORA-
DATA\ORCL\REDO_01.LOG', DBMS_LOGMNR.REMOVEFILE);
```

Una vez que se ha especificado la lista de los redo a analizar, para iniciar una sesión teclearemos:

```
SQL> EXECUTE
DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'C:\ORACLE\PRO-
DUCT\10.1.0\ORADATA\ORCL\LOGMINER\ORCLDICT.ORA');
```

Una vez ejecutado podemos consultar las siguientes vistas:

- V\$LOGMNR\_DICTIONARY: con la información del archivo de diccionario en uso.
- V\$LOGMNR\_PARAMETERS: visualiza los valores actuales de los parámetros de LogMiner.
- V\$LOGMNR\_LOGS: con la información de los redo a analizar, aquí se ven los que se han añadido con **DBMS\_LOGMNR.ADD\_LOGFILE**.
- V\$LOGMNR\_CONTENTS: esta vista contiene los resultados del análisis, es decir, el contenido del o de los Redo Log que se están analizando. Los datos de estas dos vistas se crean al iniciar una sesión de LogMiner.

Para cerrar una sesión teclearemos:

```
SQL> EXECUTE DBMS_LOGMNR.END_LOGMNR;
```

Al cerrar la sesión el contenido de las vistas V\$LOGMNR\_LOGS y V\$LOGMNR\_CONTENTS desaparece.

Para ver con más detalle el contenido de esta vista vamos a crear un usuario al que luego SYS le creará una tabla con los datos de esta vista. A continuación crearemos una conexión ODBC para poder abrir esta tabla en ACCESS; así pues, nos conectamos con SYS y hacemos lo siguiente:



## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs

- Creamos el usuario y le damos permisos;

```
SQL> CREATE USER USUPRUE IDENTIFIED BY USUPRUE TEMPORARY
TABLESPACE TEMP DEFAULT TABLESPACE USERS;
SQL> ALTER USER USUPRUE QUOTA UNLIMITED ON USERS;
SQL> GRANT CONNECT, RESOURCE TO USUPRUE;
```

- Iniciamos una sesión de LogMiner, primero indicamos el log a analizar, y luego arrancamos:

```
SQL> EXECUTE
DBMS_LOGMNR.ADD_LOGFILE('C:\ORACLE\PRODUCT\10.1.0\ORA-
DATA\ORCL\RED 002.LOG',
DBMS_LOGMNR.NEW);
SQL> EXECUTE DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'C:\ORA-
CLE\PRODUCT\10.1.0\ORADATA\ORCL\LOGMINER\ORCLDICT.ORA');
```

- A continuación, creamos la tabla en el esquema del usuario USUPRUE:

```
SQL> CREATE TABLE USUPRUE.LOGMNR_CONTENTS AS SELECT *
FROM V$LOGMNR_CONTENTS;
```

- Cerramos Log Miner: EXECUTE DBMS\_LOGMNR.END\_LOGMNR;

- Creamos un origen de datos ODBC, para ello vamos a *Inicio/Configuración/Panel de Control/Herramientas administrativas*.

Desde aquí abrimos *Orígenes de datos (ODBC)*. También se puede acceder desde el menú de Oracle 10g: *Configuration and Migration Tools / Microsoft ODBC Administrator*.

Dentro de la pestaña *DSN de Usuario*, pulsamos el botón *Agregar*, donde se muestran todos los controladores para establecer un origen de datos, hay que elegir *Oracle en OraDb10g\_home1* (ver Figura 13.7).

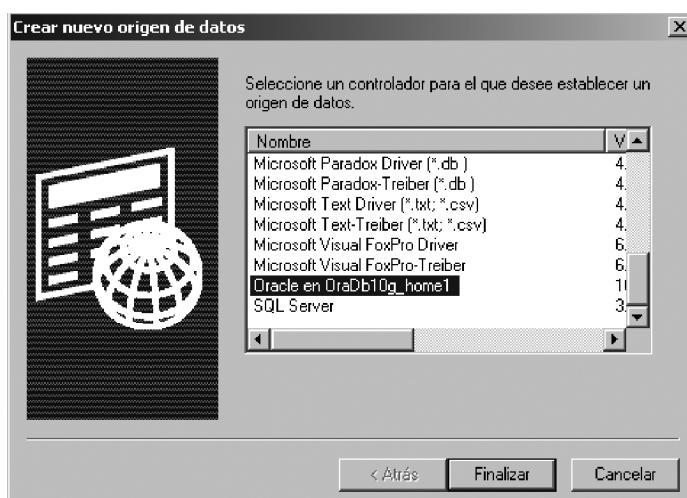


Figura 13.7. Crear un origen de datos en Oracle 10g.

## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs



- Seguidamente pulsamos el botón *Finalizar* y, a continuación, va a pedir la configuración de la conexión, en la que teclearemos el nombre del origen de datos y una descripción. Este nombre es el que luego veremos en la pestaña de DSN de usuario. En nombre del servicio elegimos la instancia ORCL, y en usuario ponemos USUPRUE. Ver Figura 13.8. Si se pulsa al botón *Test Connection* se realizará una prueba para ver si se ha configurado bien. Pulsamos *OK*, y el origen creado aparece en la lista de DSNs de usuario.

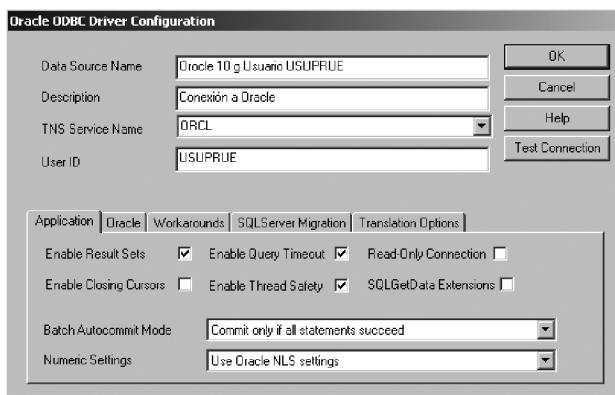


Figura 13.8. Crear un origen de datos en Oracle 10g.

- Una vez creado el origen de datos abrimos Access. Creamos una base de datos en blanco. Desde el menú *Archivo* elegimos *Obtener datos externos / Importar*. En la ventana de importar elegimos en tipo de archivo *Bases de datos ODBC* (ver Figura 13.9). A continuación, aparecerán todos los ODBC. Elegimos la pestaña *Origen de datos de equipo*, en ella debe figurar el nombre que pusimos en el punto anterior (ver Figura 13.10). Pulsamos *Aceptar*. Seguidamente pide la conexión a Oracle con el usuario USUPRUE y a la instancia o servicio ORCL. Una vez conectado aparecerán todos los objetos a los que tiene acceso ese usuario. Elegimos de la lista la tabla creada anteriormente USUPRUE.LOGMNR\_CONTENTS y pulsamos *Aceptar* (ver figura 13.11).

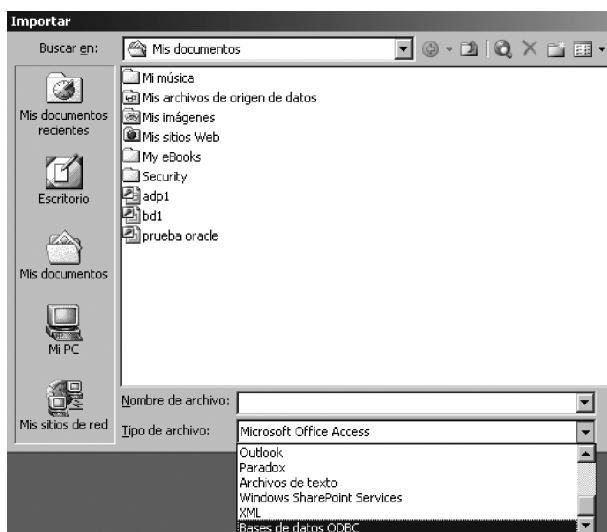


Figura 13.9. Importar datos desde ODBC en Access.



## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs

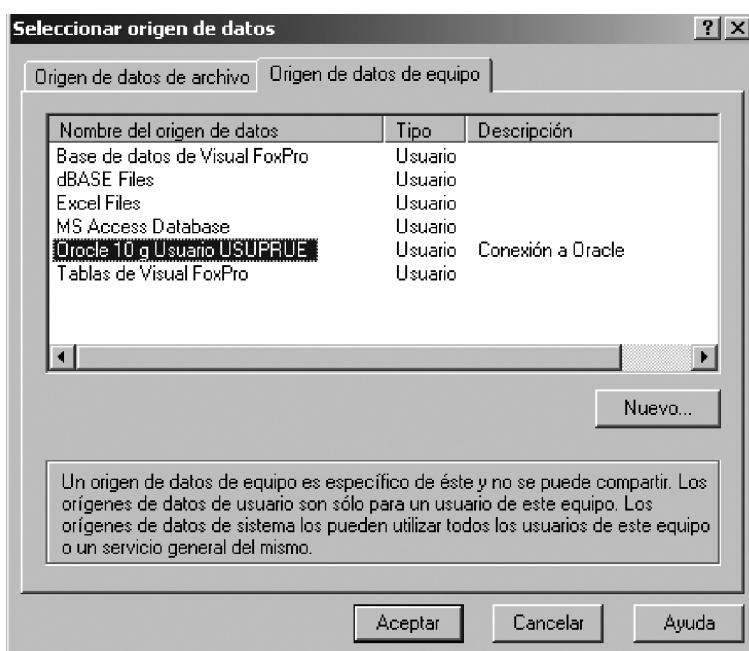


Figura 13.10. Elección del origen asociado a la instancia ORCL.

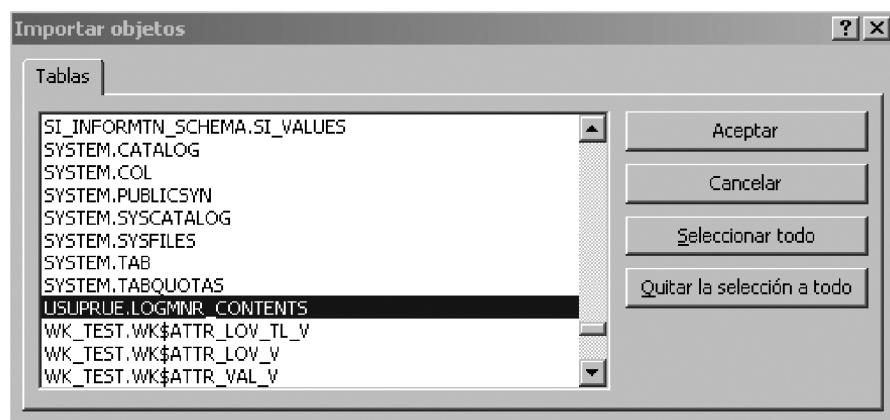


Figura 13.11. Elección del objeto a importar.

- Ya en Access podemos fijarnos en las columnas y los datos de esta tabla. En la Figura 13.12 se muestran operaciones del usuario SCOTT. Las columnas a destacar son:
  - SEG\_OWNER: el propietario que realiza la operación.
  - TABLE\_NAME y SEG\_NAME: indican el nombre del objeto.
  - TABLE\_SPACE: el *tablespace* donde se realiza la operación.
  - OPERATION: tipo de operación, INSERT, DELETE, una sentencia DDL.

## 13. Administración de Oracle II

### 13.6 Análisis de los Redo Logs

- SQL\_REDO: sentencia SQL ejecutada, aquí la columna ROLLBACK vale 1 de reconstrucción.
- SQL\_UNDO: sentencia SQL de deshacer, aquí ROLLBACK vale 0.
- TIMESTAMP: Fecha de la operación realizada.
- SCN: el número de cambio del sistema.

| SEG_OWI | SEG_NAME | TABLE_NAME | SEG_TYP | TABLE_SPA | SESS | USE | ROLLE  | OPERATIO | SQL_REDO                                        |
|---------|----------|------------|---------|-----------|------|-----|--------|----------|-------------------------------------------------|
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 1   | DELETE |          | delete from "SCOTT"."NUEEMPL" where ROWID = 'A' |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 1   | DELETE |          | delete from "SCOTT"."NUEEMPL" where ROWID = 'A' |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 1   | DELETE |          | delete from "SCOTT"."NUEEMPL" where ROWID = 'A' |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUEEMPL  | NUEEMPL    | TABLE   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUEEMPL"("EMPNO","ENAME",  |
| SCOTT   | NUFFMPI  | NUFFMPI    | TARIF   | USERS     | 0    | 0   | INSERT |          | insert into "SCOTT"."NUFFMPI"("EMPNO","ENAME",  |

Figura 13.12. Vista V\$LOGMNR\_CONTENTS desde Access, usuario SCOTT.

Para acotar y filtrar valores se puede ejecutar LogMiner a partir de un SCN, o en un rango de SCNs utilizando las opciones STARTSCN y ENDSCN:

```
SQL> EXECUTE
DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\LOGMINER\ORCLDICT.ORA', STARTSCN
=>5331831, ENDSCN=>5331839);
```

También se puede ejecutar LogMiner indicando un intervalo de tiempo utilizando las opciones STARTTIME, ENDTIME, por ejemplo, localizar todas las operaciones ocurridas el día 22/08/2005 de 10:30:00 a 11:30:00:

```
SQL> EXECUTE
DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\LOGMINER\ORCLDICT.ORA', STARTTIME
=> TO_DATE('22/08/2005 10:30:00', 'DD/MM/YYYY HH:MI:SS'), ENDTIME => TO_DATE('22/08/2005 11:30:00', 'DD/MM/YYYY
HH:MI:SS'));
```



## 13. Administración de Oracle II

### 13.7 Auditoría

## 13.7 Auditoría

Las auditorías se utilizan para realizar un seguimiento de todo lo que acontece en la base de datos, seguimiento de las instrucciones SQL y seguimiento de los objetos y todas sus operaciones. La base de datos admite la posibilidad de auditar todas las operaciones que tienen lugar en ella. Los registros de auditoría pueden escribirse tanto en la tabla **SYS.AUD\$** como en la pista de auditoría del sistema operativo. Esta segunda opción depende del sistema operativo.

Cuando se realizan auditorías, la funcionalidad de la base de datos es dejar constancia de los comandos correctos e incorrectos. Hay que intentar restringir la auditoría, definir opciones de auditoría mínimas que satisfagan los requisitos y activar las opciones de auditoría sólo cuando sea necesario. Las auditorías generan un gran número de registros, por eso, a la hora de auditar hay que intentar reducir el número de registros generados. Así la auditoría de objetos se debe utilizar siempre que sea posible. Si hay que usar auditoría de sentencias o de privilegios hay que especificar los usuarios a auditar, auditar por sesión y no por acceso, y auditar los éxitos o los fallos, pero no ambos.

Si los registros de auditoría se guardan en la tabla SYS.AUD\$, hay que pasar periódicamente sus datos a un archivo y hacer un TRUNCATE o DELETE en la tabla, pues como se almacena en el *tablespace* SYSTEM puede provocar problemas de espacio si esta tabla no se limpia. El rol **DELETE\_CATALOG\_ROLE** permite que un usuario borre los registros de esta tabla.

Si los registros se guardan en la pista de auditoría del sistema operativo hay que controlar y supervisar el aumento de la pista de auditoría. Si se llena las sentencias auditadas no se ejecutarán correctamente.

Para activar la auditoría en una base de datos hay que modificar el parámetro AUDIT\_TRAIL, cuyos valores pueden ser los siguientes:

- NONE: valor por defecto, desactiva la auditoría.
- DB: activa la auditoría y escribe todos los registros de auditoría en la tabla SYS.AUD\$, (la pista de auditoría de la base de datos).
- OS: activa la auditoría y escribe todos los registros de auditoría en la pista de auditoría del sistema operativo, si éste lo permite.

Vamos a probar la auditoría en Oracle. Para ello nos conectamos como SYS y modificamos el parámetro AUDIT\_TRAIL a DB, para que tenga vigencia hay que parar y arrancar la base de datos:

```
SQL> ALTER SYSTEM SET AUDIT_TRAIL='DB' SCOPE=SPFILE;
```

```
System altered.
```

```
SQL> SHUTDOWN IMMEDIATE
```

```
SQL> STARTUP
```

```
SQL> SHOW PARAMETER AUDIT_TRAIL
```

| NAME        | TYPE   | VALUE |
|-------------|--------|-------|
| audit_trail | string | DB    |



Para ver el resultado de las auditorías se utilizarán las siguientes vistas:

- DBA\_AUDIT\_TRAIL: contiene todas las entradas de la pista de auditoría.
- DBA\_AUDIT\_OBJECT: contiene las entradas de las auditorías sobre los objetos de los esquemas y sobre privilegios del sistema.
- DBA\_AUDIT\_SESSION: contiene las entradas de conexión y desconexión a la base de datos.

Las vistas que contienen información acerca de las opciones de auditoría activas son las siguientes:

- ALL\_DEF\_AUDIT\_OPTS: contiene las opciones de auditoría por defecto.
- DBA\_STMT\_AUDIT\_OPTS: contiene las opciones de auditoría de sentencias.
- DBA\_PRIV\_AUDIT\_OPTS: contiene las opciones de auditoría de privilegios.
- DBA\_OBJ\_AUDIT\_OPTS: contiene las opciones de auditoría sobre objetos. Por ejemplo si se audita las inserciones y los updates en la tabla NUEEMPLE de SCOTT, las columnas INS y UPD contendrán S/S:

```
SQL> AUDIT INSERT,UPDATE ON SCOTT.NUEEMPLE;
SQL> SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER=
 'SCOTT' AND
 OBJECT_NAME = 'NUEEMPLE';
```

### A. Auditorías de inicio de sesión

Esta opción audita cada intento de conectarse a la base de datos. El comando para iniciar la auditoría de los inicios de sesiones AUDIT SESSION. Para desactivar la auditoría pondremos: NOAUDIT SESSION.

**AUDIT SESSION;** Para todos los intentos de conexión.

**AUDIT SESSION WHENEVER SUCCESSFUL;** Para las conexiones correctas  
**AUDIT SESSION WHENEVER NOT SUCCESSFUL;** Para las conexiones fallidas

La vista DBA\_AUDIT\_SESSION visualizará todos los intentos de conexión y de desconexión. Las columnas más significativas son:

- OS\_USERNAME: usuario del sistema operativo.
- USERNAME: usuario de Oracle.
- TERMINAL: identificación del terminal utilizado.
- TIMESTAMP: fecha y hora de conexión.
- LOGOFF\_TIME: fecha y hora de desconexión.



## 13. Administración de Oracle II

### 13.7 Auditoría

- RETURNCODE: tipo de fallo, devuelve 0 si se ha conectado, 1005 si se introduce usuario pero no contraseña, 1017 si el usuario, o la *password* es errónea. Fíjate en los errores que devuelve Oracle al conectar un usuario, en el ejemplo el usuario ficticio no existe, y Scott sí existe pero no se ha tecleado su clave:

```
SQL> CONNECT FICTICIO/A
ERROR:
ORA-01017: invalid username/password; logon denied
SQL> CONNECT SCOTT
Enter password:
ERROR:
ORA-01005: null password given; logon denied
```

#### Caso práctico

- 12** **Audita todos los intentos de conexión y prueba conexiones con el usuario SCOTT, que sí existe en la base de datos y un usuario FICTICIO que no existe. Pruébalo desde SQLPLUS, después visualizaremos la vista DBA\_AUDIT\_SESSION. Al final cerrar la auditoría.**

```
SQL>AUDIT SESSION;
SQL> CONNECT FICTICIO/A
SQL> CONNECT SCOTT
SQL> SELECT OS_USERNAME, USERNAME, TERMINAL, TIMESTAMP, LOGOFF_TIME,
DECODE (RETURNCODE,'0','CONECTADO', '1005','NO TECLEÓ CLAVE','1017', 'USUARIO O
CLAVE ERRÓNEA', RETURNCODE) FROM DBA_AUDIT_SESSION;
SQL> NOAUDIT SESSION;
```

## B. Auditorías de sentencias y privilegios

Este tipo de auditoría contempla cualquier acción que afecte a un objeto, como puede ser: una tabla, un *tablespace*, un sinónimo, un usuario, un índice, etcétera.

Las acciones CREATE, ALTER y DROP pueden agruparse durante la auditoría de una sentencia SQL. Todos los comandos de nivel de sistema pueden auditarse, por ejemplo, para auditar todos los comandos que afectan a roles pondremos AUDIT ROLE. Esta orden auditará los comandos CREATE, ALTER, DROP y SET ROLE. Para auditar todos los comandos que afectan a usuarios pondremos AUDIT USER. Para desactivar estas opciones pondremos NOAUDIT ROLE y NOAUDIT USER.

La auditoría de privilegios audita el uso de los privilegios del sistema. Por ejemplo, si ponemos AUDIT SELECT ANY TABLE BY SCOTT BY ACCESS; (siempre que el usuario SCOTT utilice el privilegio SELECT ANY TABLE para consultar tablas de otros usuarios a las que no tiene concedidos privilegios) se generará un registro de auditoría. Cuando se audita se comprueban en primer lugar los privilegios de propietario, luego los de objeto y, a continuación, los de sistema.



Se utiliza la vista DBA\_AUDIT\_OBJECT para ver los registros de auditoría de sentencias y privilegios del sistema, y las vistas DBA\_STMT\_AUDIT\_OPTS y DBA\_PRIV\_AUDIT\_OPTS para ver las opciones de auditoría. A cada acción que puede auditarse se le asigna un código numérico dentro de la base de datos. A estos códigos se puede acceder a través de la vista AUDIT\_ACTIONS:

```
SQL> SELECT * FROM AUDIT_ACTIONS;
ACTION NAME

0 UNKNOWN
1 CREATE TABLE
2 INSERT
3 SELECT
4 CREATE CLUSTER
5 ALTER CLUSTER
6 UPDATE
7 DELETE
8 DROP CLUSTER
9 CREATE INDEX
10 DROP INDEX
```

El formato para activar las opciones de auditoría de privilegios o sentencias es el siguiente:

```
AUDIT {sentencia | privilegio_de_sistema}[,{sentencia |
privilegio_de_sistema}]...
[BY usuario [, usuario]...]
[BY {SESSION | ACCESS}]
[WHENEVER [NOT] SUCCESSFUL]
```

- BY {SESSION | ACCESS} indica que el registro de auditoría se escribe una sola vez por sesión (éste es el valor por defecto) o cada vez que se haga un acceso a un objeto (en el caso de DDL se audita por acceso). Audit BY ACCESS genera gran número de registros de auditoría. Esta opción se utiliza de manera limitada, por ejemplo, para controlar las acciones durante un intervalo de tiempo.
  - WHENEVER: especifica que la auditoría se lleva a cabo cuando se completen las sentencias SQL de forma correcta o incorrecta, el valor por defecto incluye las dos opciones.

## Caso práctico



- 13 Nos conectamos con SYS para auditar el privilegio CREATE TABLE a SCOTT. Consultamos la vista DBA\_PRIV\_AUDIT\_OPTS para ver si aparece esta auditoría. Borraremos las filas de AUD\$ para ver sólo esta auditoría:

```
SQL> AUDIT CREATE TABLE BY SCOTT;
SQL> DELETE FROM AUD$;
```

**(Continúa)**



## 13. Administración de Oracle II

### 13.7 Auditoría

(Continuación)

```
SQL> SELECT * FROM DBA_PRIV_AUDIT_OPTS;
```

| USER_NAME | PROXY_NAME | PRIVILEGE    | SUCCESS   | FAILURE   |
|-----------|------------|--------------|-----------|-----------|
| SCOTT     |            | CREATE TABLE | BY ACCESS | BY ACCESS |

Luego nos conectamos con SCOTT y creamos una tabla. Seguidamente, desde SYS consultamos la vista DBA\_AUDIT\_OBJECT:

```
SQL> CREATE TABLE PRUEBA (V1 NUMBER(3), V2 NUMBER(3));
SQL> SELECT TERMINAL, OWNER, USERNAME, OBJ_NAME, TO_CHAR(TIMESTAMP, 'DD/MON/YY
HH24:MI:SS'), RETURNCODE, ACTION_NAME FROM DBA_AUDIT_OBJECT;
```

### C. Auditorías de objetos

También se pueden auditar acciones de manipulación de datos sobre objetos. Incluyen las operaciones de SELECT, INSERT, UPDATE y DELETE sobre tablas. Se utilizan las vistas DBA\_AUDIT\_OBJECT para ver los registros de auditoría de objetos. El formato que se utiliza es similar al anterior:

```
AUDIT sentencia [,sentencia]...
ON { [esquema.]objeto | DEFAULT}
[BY {SESSION | ACCESS}]
[WHENEVER [NOT] SUCCESSFUL]
```



#### Caso práctico

##### 14 Nos conectamos con SYS y auditamos todos los INSERT que se realicen en la tabla NUEEMPL de SCOTT:

```
SQL> AUDIT INSERT ON SCOTT.NUEEMPL;
```

- Abrimos una sesión con SCOTT e insertamos dos registros:

```
SQL> INSERT INTO NUEEMPL VALUES (111,'PEPITO','ENFERMERO', 7839,'09/05/80',
2000.34,100.93,10);
SQL> INSERT INTO NUEEMPL VALUES (222,'MARÍA','MÉDICO', 7839,'10/07/83',
2500.98,170.88,20);
SQL> COMMIT;
```

- Desde SYS comprobamos lo que ha ocurrido, y lo que ha hecho SCOTT, consultamos la hora y la acción realizada, utilizamos la vista DBA\_AUDIT\_OBJECT. USERNAME es el usuario que realiza la acción, y OWNER el propietario del objeto:

```
SQL> SELECT TERMINAL, OWNER, OBJ_NAME, TO_CHAR(TIMESTAMP, 'DD/MON/YY
HH24:MI:SS'), ACTION_NAME FROM DBA_AUDIT_OBJECT WHERE USERNAME='SCOTT';
```

(Continúa)

## 13. Administración de Oracle II

13.7 Auditoría



(Continuación)

Vemos que aparece un registro en la vista en la que se indica que SCOTT ha realizado un nombre de acción SESSION REC, aunque hemos insertado dos registros, sólo aparece una vez en la vista pues la auditoría se realiza por SESSION, si no se especifica BY ACCESS.

- Ahora anulamos esta auditoría y SYS audita esta misma tabla por acceso.

```
SQL> NOAUDIT INSERT ON SCOTT.NUEEMPL;
SQL> AUDIT INSERT ON SCOTT.NUEEMPL BY ACCESS;
```

- Abrimos de nuevo (no vale la sesión anterior) una sesión con SCOTT e insertamos dos registros:

```
SQL> INSERT INTO NUEEMPL VALUES (333,'JUANITO','ANALISTA', 7839,'11/11/84',
2400.34,180.93,20);
SQL> INSERT INTO NUEEMPL VALUES (444,'FELISA','TÉCNICO', 7839,'10/10/82',
2300.08,160.68,20);
SQL> COMMIT;
```

- Y desde SYS consultamos la vista DBA\_AUDIT\_OBJECT. Observa que ahora aparecen dos registros más y que ACTION\_NAME es INSERT. Fíjate en la hora de inserción.
- Ahora SYS crea un usuario al que va a dar permiso de INSERT sobre la tabla NUEEMPL de SCOTT. Nos conectaremos con este usuario e insertaremos dos registros. Luego consultamos la vista DBA\_AUDIT\_OBJECT.

```
SQL> CREATE USER PRUE_AU IDENTIFIED BY A
TEMPORARY TABLESPACE TEMP
DEFAULT TABLESPACE USERS;
SQL> ALTER USER PRUE_AU QUOTA UNLIMITED ON USERS;
SQL> GRANT CONNECT TO PRUE_AU;
SQL> GRANT INSERT ON SCOTT.NUEEMPL TO PRUE_AU;
```

- Abrimos una sesión con PRUE\_AU e insertamos dos registros, uno erróneo y el otro correcto. Por ejemplo el día de la fecha del primer registro la ponemos mal, el error es ORA-01847:

```
SQL> INSERT INTO SCOTT.NUEEMPL VALUES (555,'ALICIA','PROFESORA', 7839,'1111/05/70'
,2000.04,140.93,20);
SQL> INSERT INTO SCOTT.NUEEMPL VALUES (666,'JOSÉ','INGENIERO', 7839,'12/07/82'
,2600.98,170.88,30);
SQL> COMMIT;
```

- De nuevo desde SYS comprobamos lo ocurrido. En este caso, vemos los objetos propiedad de SCOTT. Fíjate que en la columna USERNAME aparecerá PRUE\_AU. Observa que si se realiza un INSERT fallido, RETURNCODE devolverá el código de error, si es satisfactorio devuelve 0:

```
SQL> SELECT TERMINAL, OWNER, USERNAME, OBJ_NAME, TO_CHAR(TIMESTAMP, 'DD/MON/YY
HH24:MI:SS'), RETURNCODE, ACTION_NAME FROM DBA_AUDIT_OBJECT WHERE OWNER='SCOTT';
```



## 13. Administración de Oracle II

### Conceptos básicos



Hay dos tipos de copias de seguridad:

- **Físicas:** son copias de los datos de la base de datos. Se pueden realizar con utilidades del sistema operativo o con la utilidad RMAN de Oracle.
- **Lógicas:** son copias de distintas partes de una base de datos (*tablespaces*, tablas, esquemas, etcétera) extraídos con órdenes Oracle (EXPORT - IMPORT) y almacenados en archivos binarios. Se pueden utilizar en complemento de las copias físicas.

Las copias de seguridad de base de datos entera copian todos los archivos de datos y de control, y se pueden realizar con la base de datos cerrada o abierta. Dependiendo del momento en el que la copia de seguridad se realiza, se pueden distinguir dos tipos:

- **Copias consistentes o en frío:** realizadas cuando la base de datos está cerrada, con las opciones NORMAL o IMMEDIATE, ésta contiene las últimas modificaciones. En esta copia todas las cabeceras de los archivos de la base de datos son consistentes con el archivo de control, y cuando se restauran los archivos, la base de datos puede abrirse sin recuperación. En modo NOARCHIVELOG, sólo

es válida la copia de seguridad entera consistente para la restauración y la recuperación.

- **Copias inconsistentes o en caliente:** realizadas cuando la base de datos esta en uso, o realizadas después de un cierre anormal. Este tipo se ejecuta cuando la base de datos tiene que estar disponible las 24 horas del día. En este caso las cabeceras de los archivos de datos con el de control son inconsistentes, para que la base de datos vuelva a ser consistente se necesita una recuperación. Este tipo de copia sólo se debe realizar en bases de datos en modo ARCHIVELOG. Ver Figura 13.13

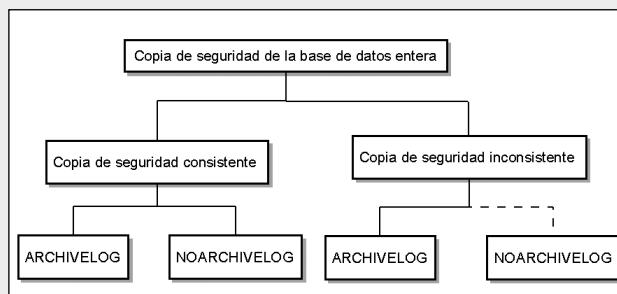


Figura 13.13. Copia de seguridad de la base de datos entera.



## Actividades complementarias



- 1** Crea un *tablespace* de nombre **COMPRAS** asociándole un archivo en disco llamado '**COMPRAS.ORA**' de 5 Megabytes.
- 2** Modifica el *tablespace* del ejercicio 1 para que pueda auto extenderse automáticamente, sin límite de espacio en disco.
- 3** Realiza un análisis de los Redo Log en línea utilizando LogMiner.
- 4** Realiza una copia de seguridad de todos los *tablespaces*, con la base de datos abierta.
- 5** Exporta el esquema completo de un usuario de la base de datos, e importarlo en otra base de datos.
- 6** Crea una tabla y, seguidamente, crea una secuencia que genere números empezando en 10 y con incremento de 10. Inserta filas en la tabla creada utilizando la secuencia que se ha creado en alguna de las columnas.
- 7** Realiza una auditoría para controlar todos los intentos de conexión con la base de datos. Importa la tabla de auditoría AUD\$ a Access para observar las operaciones realizadas.
- 8** Relaciona las vistas V\$ con la información que contienen:
- V\$DATAFILE
  - V\$DATABASE
  - V\$LOG
  - V\$LOGFILE
- 9** Al ejecutar la orden **ALTER SYSTEM SWITCH LOGFILE**:
- Se produce un cambio de log manual si la base de datos está en NO ARCHIVELOG.
  - Se realiza la copia del Redo Log online si la base de datos está en ARCHIVELOG, y además se produce un cambio de log manual.
  - Esta orden sólo se puede ejecutar si la base de datos está en modo ARCHIVELOG.
  - Se hace una copia de seguridad de todos los Redo Log.
- 10** Una copia de seguridad incremental:
- Contiene los cambios producidos desde la última copia incremental del mismo nivel.
  - Contiene los cambios producidos desde la última copia incremental del mismo o menor nivel.
  - Contiene los cambios producidos desde la copia incremental de nivel 0.
  - Oracle sólo permite copias completas y acumulativas.









## 12. Administración de Oracle I

### 12.5 Gestión de seguridad

(Continuación)

| RECURSO               | FUNCIÓN                                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| PRIVATE_SGA           | Indica la cantidad de espacio privado que una sesión puede reservar en el área SQL compartida de la SGA (para la opción servidor compartido). |
| COMPOSITE_LIMIT       | Indica un límite compuesto basado en los límites anteriores.                                                                                  |
| FAILED_LOGIN_ATTEMPTS | Número de intentos de acceso sin éxito consecutivos que producirá el bloqueo de la cuenta.                                                    |
| PASSWORD_LIFE_TIME    | Número de días que puede utilizarse una contraseña antes de que caduque.                                                                      |
| PASSWORD_REUSE_TIME   | Número de días que deben pasar antes de que se pueda reutilizar una contraseña.                                                               |
| PASSWORD_REUSE_MAX    | Número de veces que debe cambiarse una contraseña antes de poder reutilizarla.                                                                |
| PASSWORD_LOCK_TIME    | Número de días que quedará bloqueada una cuenta si se sobrepasa el valor del parámetro FAILED_LOGIN_ATTEMPTS.                                 |
| PASSWORD_GRACE_TIME   | La duración en días del periodo de gracia durante el cual una contraseña puede cambiarse cuando ha alcanzado su valor PASSWORD_LIFE_TIME.     |

Para activar el uso de perfiles en el sistema, el administrador ha de ejecutar esta orden: **ALTER SYSTEM SET RESOURCE\_LIMIT=TRUE;** (FALSE desactiva la utilización de perfiles). Para asignar un perfil a un usuario se puede utilizar la orden **ALTER USER: ALTER USER USUARIO PROFILE nombreperfil;** O bien al crear el usuario se le puede asignar un perfil.



### Caso práctico

- 8 Por razones de seguridad, creamos el perfil **PERFIL1**, en el que limitamos a uno el número de sesiones concurrentes por usuario y a dos minutos el tiempo de conexión permitido por sesión:

```
CREATE PROFILE PERFIL1 LIMIT SESSIONS_PER_USER 1 CONNECT_TIME 2;
```

Dado que los demás límites de recurso no se mencionan en la instrucción CREATE PROFILE, se utilizarán los valores asignados por defecto por el sistema.

A continuación, se crea un usuario llamado PRUEBA. Se le asigna este perfil y se le concede el rol CONNECT:

```
CREATE USER PRUEBA IDENTIFIED BY PRUEBAQUOTA 100K ON USERS PROFILE PERFIL1;
GRANT CONNECT TO PRUEBA;
```

(Continúa)