



9.1 Introducción

La tarea de administración de una base de datos para varios usuarios es bastante compleja, por lo que se suele encomendar a una o varias personas. El administrador de la base de datos (*DBA: Data Base Administrator*) suele ser un usuario muy experimentado capaz de enfrentarse a los problemas de los demás usuarios y a los fallos que el sistema va planteando.

Son tareas del administrador de Oracle: instalar Oracle, diseñar y crear una base de datos, arrancar y detener la base de datos, crear y controlar usuarios, conceder privilegios, gestionar el espacio, hacer copias de seguridad y recuperar la base de datos.

9.2 ¿Qué es Oracle 10g?

Oracle 10g es la nueva suite de productos software de la compañía Oracle, basados todos ellos en la tecnología *Grid Computing* (o *computación Grid*).

El **Grid** es una nueva arquitectura que agrupa múltiples servidores y recursos de almacenamiento y procesamiento en una estructura más económica y flexible que atiende a todas las necesidades de la organización y donde los recursos para el procesamiento de datos están disponibles para los usuarios, según los vayan necesitando.

Los tres elementos sobre los que se articula *Oracle Grid Computing* son: *Oracle Database 10g*, *Oracle Application Server 10g* y *Oracle Enterprise Manager 10g*:

- *Oracle Database 10g*. Es el motor de la base de datos. Dispone de herramientas capaces de gestionar eficazmente el almacenamiento de la información, utilizar de forma óptima los recursos, ofrecer un máximo nivel de atención en los servicios ofrecidos, etcétera. Gestiona de forma eficaz toda la información relacional, correo electrónico, documentos, multimedia, XML y espacial.
- *Oracle Application Server 10g*. Es el servidor de aplicaciones Oracle. Como nivel intermedio de una arquitectura de tres niveles consiste en un conjunto de servicios que permite a las aplicaciones basadas en exploradores web interactuar con las bases de datos de Oracle.
- *Oracle Enterprise Manager 10g*. Proporciona un marco adecuado para llevar a cabo las tareas de administración de la base de datos. Se accede a través del navegador web mediante la URL http://nombre_del_host:5500/em. Se obtiene una pantalla desde la que hay que escribir el nombre de usuario, la clave y el rol administrativo. Al instalar Oracle Database 10g, se instala este elemento.



9. Administración de Oracle

9.3 Arquitectura Oracle

9.3 Arquitectura Oracle

Para comprender mejor una base de datos necesitamos conocer su arquitectura. Cuando hablamos de una «base de datos» no sólo nos estamos refiriendo a los datos físicos, sino también a la combinación de objetos físicos, de memoria y de proceso que se describen a continuación.

A. Componentes de la base de datos

Los componentes de una base de datos Oracle son: *archivos de datos (database files)*, *archivos de diario o de transacciones (log files)* y *archivos de control (control files)*.

- **Archivos de datos (database files):** Contienen toda la información de la base de datos: datos de usuario y datos de sistema. Antes de introducir información en la base de datos, es necesario crear un espacio donde ubicaremos las tablas (*tablespaces*) que contendrán los datos. Los *tablespaces* nos ayudan a organizar la información contenida en la base de datos; así, podemos tener un *tablespace* para almacenar los datos de la aplicación de almacén, otro para almacenar los datos de la aplicación de nóminas, etcétera.

Cada *tablespace* consta de uno o más archivos en disco. Un archivo de datos sólo puede pertenecer a un único *tablespace*. Al instalar Oracle se crean varios *tablespaces*. Algunos son:

- **SYSTEM.** En él se almacena toda la información que Oracle necesita para gestionarse a sí misma, por ejemplo: el diccionario de datos. Se almacena en el archivo SYSTEM01.DBF.
- **USERS.** Contiene información personal de los usuarios. Normalmente, es el lugar en el que el DBA nos deja almacenar las tablas para realizar pruebas. Se almacena en el archivo USERS01.DBF.
- **TEMP.** Aquí Oracle almacena las tablas temporales (para gestionar sus transacciones). Se almacena en el archivo TEMP01.DBF.
- **UNDOTBS1.** Es el *tablespace* donde Oracle guarda la información de deshacer. Se utiliza para almacenar la imagen anterior de los datos antes de permitir actualizaciones. Esto permite recuperar los datos cuando no se completa una transacción. Se almacena en el archivo UNDOTBS01.DBF.
- **Registros de rehacer o Redo_Log: el registro de las transacciones:** Se trata de archivos de datos en los que Oracle registra todas las transacciones o modificaciones (INSERT, UPDATE y DELETE) que se producen. Esto permite recuperar la base de datos si hay problemas. Suele haber varios registros de rehacer almacenados físicamente en los archivos RED001.LOG, RED002.LOG y RED003.LOG.

Un registro de Redo_Log contiene: identificación de la transacción, dirección de bloque, número de fila, número de columna y valor anterior y nuevo del dato modificado.



- **Archivos de control (control files):** Contienen información sobre los archivos asociados con una base de datos Oracle. Todas las modificaciones importantes que se hagan en la estructura de la base de datos se registran en el archivo de control, manteniendo así la integridad. Oracle recomienda que la base de datos tenga un mínimo de dos archivos de control en discos diferentes. Si se daña un disco de control, se podría restaurar utilizando la copia intacta. Los archivos de control que se crean en la instalación son: CONTROL01.CTL, CONTROL02.CTL y CONTROL03.CTL, y contienen la siguiente información:

- Información de arranque y parada.
- Nombre de los archivos de la base de datos y de Redo_Log.
- Información sobre *checkpoints* (puntos de control que se dan cuando se llena el Redo_Log, cuando se detiene la base de datos, etcétera).
- Fecha de creación y nombre de la base de datos.
- Estado *on-line* y *off-line* de los archivos.

B. Estructura de la memoria

Los procesos del usuario (cliente) y del servidor se comunican consigo mismos y entre ellos por medio de estructuras de memoria. Oracle utiliza dos tipos de estructuras de memoria: el **SGA** (*System Global Area*) y el **PGA** (*Program Global Area*).

- **Área global del sistema (SGA)**

Es un grupo de estructuras de memoria que sirven para almacenar los datos de la base de datos que se han consultado más recientemente. Se descompone en las siguientes zonas:

- **Conjunto compartido:** Formado por la *caché de diccionario de datos*, la *caché de biblioteca* y el *área SQL compartida*. La **caché de diccionario de datos** contiene información acerca de los datos y su estructura (por ejemplo, si un usuario puede acceder o no a una tabla). La **caché de biblioteca** contiene información sobre las instrucciones SQL ejecutadas sobre la base de datos. El **área SQL compartida** contiene el plan de ejecución y el árbol de análisis de las sentencias SQL que se ejecutan. La segunda vez que un usuario ejecuta una sentencia idéntica SQL ya ejecutada puede aprovecharse del análisis disponible en este área para acelerar su ejecución.
- **Caché de buffers:** contiene copias de los bloques de datos leídos de los archivos de datos. Los usuarios acceden a los datos en esta zona de la memoria. También se le conoce como *buffer del bloque de datos*.
- **Conjunto grande:** es un área de memoria opcional. Se define cuando se utiliza la opción de servidor compartido o cuando se realizan con frecuencia operaciones de copias de seguridad o restauración.
- **Conjunto Java:** especifica el tamaño para satisfacer los requisitos de análisis de los comandos Java.



9. Administración de Oracle

9.3 Arquitectura Oracle

Cuando un usuario se conecta a la base de datos, por ejemplo desde SQL*Plus, se crea un proceso de usuario. Oracle comprueba y valida ese usuario e inmediatamente la asigna un proceso de servidor que lleva asociado su PGA.

- **Buffer del registro de rehacer (*Redo Log buffer*):** en este área se registran las transacciones (INSERT, UPDATE, DELETE) o cambios en la base de datos antes de escribirse en los archivos de registro de rehacer.

• Área global del programa (PGA)

Es la zona de memoria utilizada por un único proceso de usuario de Oracle, y contiene datos e información del proceso. La memoria de la PGA no se comparte.

C. Procesos de soporte de la base de datos

Las relaciones entre las estructuras físicas y de memoria de la base de datos se mantienen y aplican mediante una serie de procesos soporte (de segundo plano, *background* o de fondo). Dependiendo de la configuración de la base de datos, el número de procesos varía. La gestión de éstos procesos la realiza la base de datos con poco trabajo administrativo.

Hay un conjunto de procesos del servidor que ayudan a la base de datos a funcionar: son los **procesos soporte o de fondo**. Véase la Figura 9.1.

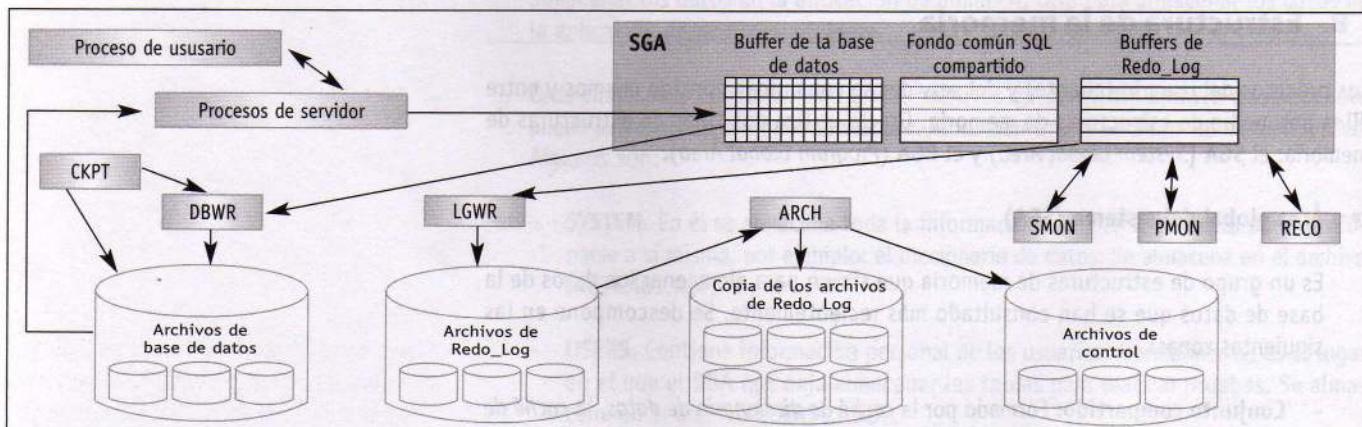


Figura 9.1. Procesos y áreas de memoria Oracle.

Son los siguientes:

- **Escritor de bases de datos DBWR (DataBase WRiter):** Este proceso es el responsable de gestionar el contenido del buffer de datos de la SGA. Lee los bloques de los archivos de datos, los almacena en la SGA y realiza escrituras de los bloques modificados en los archivos de datos. Cuando un usuario solicita una petición cuyos datos no están en el buffer de datos, el DBWR se encargará de llevar al buffer un nuevo bloque de información del archivo de datos.
- **Punto de comprobación o control CKPT (CheckPoint):** Estos puntos provocan que el DBWR escriba en los archivos de datos todos los bloques que se hayan modificado desde el último punto de control y que actualice las cabeceras de los archivos de datos y los archivos de control para registrar el punto de control. Se producen de forma automática cuando se llena un archivo de registro de rehacer Redo_Log.

9. Administración de Oracle

9.3 Arquitectura Oracle



- Escriptor de registros LGWR (Log WRiter):** Gestiona la escritura del contenido del buffer del registro de rehacer de la SGA a los archivos de Redo_Log. Es el único proceso que escribe en los archivos de registro de rehacer y el único que lee los buffers de este registro.

El proceso LGWR escribe en los archivos de Redo_Log de forma cíclica.

Esta operación permite que Oracle pueda recuperarse en cualquier momento si hay fallos. El proceso DBWR debe esperar al escritor de registros antes de escribir los bloques modificados desde los buffers del bloque de datos a los archivos de datos; es decir, en primer lugar se escribe la transacción en los registros de rehacer y luego se escribe en la base de datos.

- Supervisor del sistema SMON (System MONitor):** Es un proceso obligatorio que se ocupa de todas las recuperaciones que sean precisas durante el arranque de la base de datos. La limpia eliminando datos de las transacciones que el sistema ya no necesita y compacta los huecos libres en los ficheros de datos. Se activa de forma periódica para comprobar si su intervención es precisa.
- Supervisor de proceso PMON (Process MONitor):** Realiza una limpieza al terminar la ejecución de los procesos. Restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloques y los recursos de la SGA. Al igual que SMON, se activa de forma periódica para comprobar si es necesaria su intervención.
- Archivador ARCH (ARCHiver):** Es opcional y archiva en disco o cinta una copia de los Redo_Log cuando están llenos para una posible recuperación por fallo de disco. Se ejecuta en modo ARCHIVELOG.
- Recuperador RECO (RECOVerer):** Es opcional. Recupera transacciones distribuidas dudosas; se usa en bases de datos Oracle distribuidas.

Cuando Oracle se ejecuta en modo ARCHIVELOG, la base de datos realiza una copia de los archivos de Redo_Log antes de sobreescribirlos.

RESUMEN

La base de datos Oracle más elemental consta de:

- Uno o más archivos de datos.
- Uno o más archivos de control.
- Dos o más archivos de Redo_Log.

Internamente, esta base de datos contiene:

- Varios usuarios/esquemas.
- Uno o más segmentos de rollback.
- Uno o más espacios de tablas.
- Tablas del diccionario de datos.
- Objetos de usuarios (tablas, vistas, etcétera).

El servidor que accede a esta base de datos consta, como mínimo, de:

- Un SGA (que contiene caché del buffer del bloque de datos, caché del buffer del registro de rehacer, el fondo común SQL).
- El proceso SMON.
- El proceso PMON.
- El proceso DBWR.
- El proceso LGWR.
- Procesos de usuario con PGA asociada.

Los **segmentos** son estructuras lógicas de almacenamiento donde se alojan los objetos de la base de datos. Los segmentos de Rollback permiten efectuar la restauración de las transacciones no validadas y aseguran la consistencia en lectura.

Veamos ahora mediante un caso práctico qué hace Oracle cuando hacemos una consulta a la base de datos y cuando hacemos una actualización de una fila.



9. Administración de Oracle

9.3 Arquitectura Oracle



Caso práctico

- 1 Queremos consultar la nota de la alumna "Luisa Martínez" en la asignatura de "Informática". Para ello, hacemos la siguiente consulta a la tabla NOTAS_ALUMNOS de la base de datos a través de la siguiente sentencia SQL:

```
SELECT NOTA FROM NOTAS_ALUMNOS WHERE ASIGNATURA = 'Informática'  
AND NOMBRE_ALUMNO='Luisa Martínez';
```

La sentencia SQL es un proceso de usuario que sigue estos pasos:

1. El proceso de usuario pasa la sentencia SQL al proceso del servidor por medio del SGA.
2. Los procesos del servidor buscan en el fondo común SQL una versión ejecutable de la sentencia. Si la encuentran, se ejecuta la sentencia SQL; si no se encuentra, se procesa y se lleva su versión ejecutable al fondo común.
3. Si los datos requeridos no están en el buffer del bloque de datos, por medio del proceso **DBWR** se leen los datos de los archivos de datos y se colocan en los buffers del bloque de datos del **SGA**.
4. Una vez que los datos se encuentran en los buffers del bloque de datos, el proceso de usuario puede leer la nota y mandársela al usuario.

Éste es el ejemplo más sencillo, pues no se ha hecho ninguna modificación de los datos.

Suponemos que cambiamos la nota de Informática de Luisa Martínez. Se modifica la columna NOTA de la tabla NOTAS_ALUMNOS con el valor 7: UPDATE NOTAS_ALUMNOS SET NOTA = 7 WHERE ASIGNATURA = 'Informática' AND NOMBRE_ALUMNO = 'Luisa Martínez';

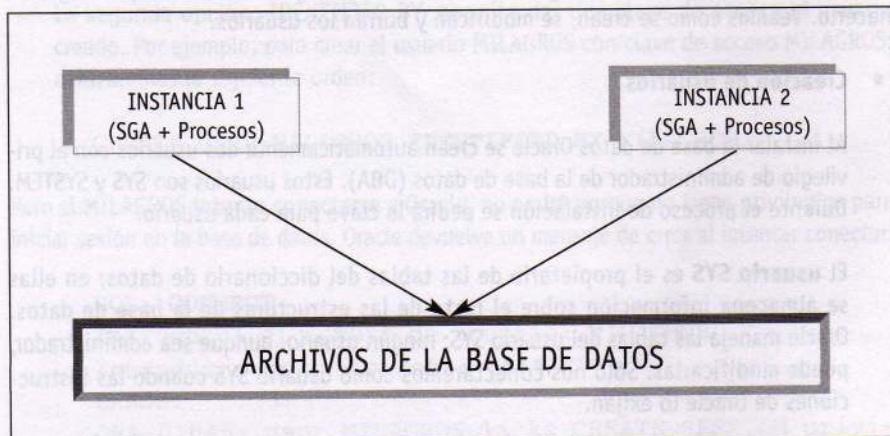
Los pasos que da Oracle con la sentencia UPDATE (actualizar) son los siguientes:

1. El proceso de usuario pasa la sentencia SQL al proceso del servidor.
2. Los procesos del servidor buscan en el fondo común SQL una versión ejecutable de la sentencia. Si la encuentran, se ejecuta la sentencia SQL; si no la encuentran, se procesa y se lleva su versión ejecutable al fondo común.
3. Si los datos requeridos no están en el buffer del bloque de datos, por medio del proceso **DBWR** se leen los datos de los archivos de datos y se colocan en los buffers del bloque de datos del **SGA**.
4. Se registra el valor antiguo de los datos en un segmento de ROLLBACK (la nota antigua era 5).
5. Se crea una transacción en el buffer de Redo_Log.
6. Se modifican los datos en los buffers del bloque de datos para reflejar la nueva nota (7).
7. Cuando hacemos un **COMMIT**, el **LGWR** escribe los buffers de Redo_Log a los archivos de Redo_Log. Se libera la información de deshacer en el segmento de rollback.
8. En algún momento se producirá un punto de comprobación o control **CKPT**, y el **DBWR** escribirá en los archivos de datos todos los bloques que se hayan modificado desde el último punto de control.



D. ¿Qué es una instancia Oracle?

Una **instancia Oracle** es un conjunto de estructuras de memoria (SGA) y procesos en segundo plano que acceden a un conjunto de archivos de base de datos (véase la Figura 9.2).



Una base de datos Oracle no está disponible a los usuarios hasta que el administrador haya iniciado la instancia y abierto la base de datos.

Cada vez que se inicia una instancia, Oracle utiliza un archivo de parámetros para asignar el SGA e iniciar los procesos en segundo plano.

Figura 9.2. Instancias y archivos de datos de Oracle.

9.4 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 el usuario puede acceder a ese objeto y qué tipo de operación puede hacer con él (SELECT, INSERT, etcétera).

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, 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.



9. Administración de Oracle

9.4 Gestión de seguridad

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 éste 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 modificarlas. 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 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 DDICTIONARY, 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];

9. Administración de Oracle

9.4 Gestión de seguridad



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 (DEFAULT). 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

- 2 La siguiente orden crea un usuario de nombre USUARIO1. 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 USUARIO1 IDENTIFIED BY USUARIO1  
DEFAULT TABLESPACE TRABAJO  
TEMPORARY TABLESPACE TEMPORAL  
QUOTA 1M ON TRABAJOS QUOTA 500K ON TEMPORAL;
```



9. Administración de Oracle

9.4 Gestión de seguridad

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.

Cada usuario puede cambiar únicamente su clave de acceso, no puede cambiar el *tablespace* por defecto, ni la cuota en los *tablespaces* ni el perfil, a no ser que tenga el privilegio ALTER USER. Por ejemplo, el USUARIO1 cambia su clave:

```
ALTER USER USUARIO1 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 USUARIO1: **DROP USER USUARIO1;** Si el usuario tiene objetos creados se visualiza un mensaje de error: ORA-01922: se debe especificar CASCADE para borrar USUARIO1 y no nos deja borrarlo, tenemos que añadir la opción CASCADE: **DROP USER USUARIO1 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.

9. Administración de Oracle

9.4 Gestión de seguridad



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 procedimientos:

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



9. Administración de Oracle

9.4 Gestión de seguridad

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.



Caso práctico

3 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;`

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 privilegios 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 la opción 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;
```

9. Administración de Oracle

9.4 Gestión de seguridad



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.

• Privilegios del sistema

Los privilegios de 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.



9. Administración de Oracle

9.4 Gestión de seguridad

PRIVILEGIO DEL SISTEMA

CREATE SYNONYM
CREATE PUBLIC SYNONYM
DROP PUBLIC SYNONYM
CREATE ANY SYNONYM
DROP ANY SYNONYM

CREATE TABLE
CREATE ANY TABLE
ALTER ANY TABLE
DROP ANY TABLE
LOCK ANY TABLE
SELECT ANY TABLE
INSERT ANY TABLE
UPDATE ANY TABLE
DELETE ANY TABLE

CREATE TABLESPACE
ALTER TABLESPACE
MANAGE TABLESPACES
DROP TABLESPACE
UNLIMITED TABLESPACE

CREATE TYPE
CREATE ANY TYPE
ALTER ANY TYPE
DROP ANY TYPE
EXECUTE ANY TYPE
UNDER ANY TYPE

CREATE USER

ALTER USER

DROP USER

CREATE VIEW
CREATE ANY VIEW
DROP ANY VIEW

SYSDBA

SYSOPER

OPERACIONES AUTORIZADAS

SYNONYM

Crear sinónimos en nuestro esquema.
Crear sinónimos públicos.
Borrar sinónimos públicos.
Crear sinónimos en cualquier esquema.
Borrar sinónimos de cualquier esquema.

TABLE

Crear tablas en nuestro esquema y generar índices sobre las tablas del esquema.
Crear una tabla en cualquier esquema.
Modificar una tabla en cualquier esquema.
Borrar una tabla en cualquier esquema.
Bloquear una tabla en cualquier esquema.
Hacer SELECT en cualquier tabla.
Insertar filas en cualquier tabla.
Modificar filas en cualquier tabla.
Borrar filas de cualquier tabla.

TABLESPACES

Crear espacios de tablas.
Modificar tablespaces.
Poner on-line u off-line a cualquier tablespace.
Eliminar tablespaces.
Utilizar cualquier espacio de cualquier tablespace.

TYPE

Crea tipos de objeto y cuerpos de tipos de objeto en el propio esquema.
Crea tipos de objeto y cuerpos de tipos de objeto en cualquier esquema.
Modifica tipos de objeto en cualquier esquema.
Elimina tipos de objeto y cuerpos de tipos de objeto en cualquier esquema.
Utiliza y hace referencia a tipos de objeto y tipos de colección en cualquier esquema.
Crea subtipos a partir de cualquier tipo de objeto no final.

USER

Crear usuarios y crear cuotas sobre cualquier espacio de tablas, establecer espacios de tablas por omisión y temporales.
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.
Eliminar usuarios.

VIEW

Crear vistas en el esquema propio.
Crear vistas en cualquier esquema.
Borrar vistas en cualquier esquema.

OTROS

Ejecutar operaciones STARTUP y SHUTDOWN, ALTER DATABASE, CREATE DATABASE, ARCHIVELOG y RECOVERY, etcétera.
Ejecutar operaciones STARTUP y SHUTDOWN, ALTER DATABASE, ARCHIVELOG y RECOVERY, etcétera.

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];
```

9. Administración de Oracle

9.4 Gestión de seguridad



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

- 4 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;
```

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}] ...;
```



9. Administración de Oracle

9.4 Gestión de seguridad



Caso práctico

5 MILAGROS retira los privilegios SELECT y UPDATE sobre TABLA1 a FRANCISCO:

```
REVOKE SELECT, UPDATE ON TABLA1 FROM FRANCISCO;
```

REVOKE ALL elimina todos los privilegios concedidos anteriormente sobre algún objeto. La opción WITH GRANT OPTION desaparece con el privilegio que le fue asignado. 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;
```

• 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_REC'D: 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_REC'D: concesiones sobre columnas recibidas por el usuario.

9. Administración de Oracle

9.4 Gestión de seguridad



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. Después conéctate con ese usuario y consulta los privilegios del sistema que tiene. Crea una tabla y 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;
```



9. Administración de Oracle

9.4 Gestión de seguridad



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.

• 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. Por ejemplo, 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 el privilegio INSERT del rol ACCESO en la table 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;

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

9. Administración de Oracle

9.4 Gestión de seguridad



Continuación

• 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.

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.



9. Administración de Oracle

9.4 Gestión de seguridad

RECURSO

FUNCIÓN

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).
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 que caducue.
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

- 6 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 PRUEBA
QUOTA 100K ON USERS PROFILE PERFIL1;
GRANT CONNECT TO PRUEBA;
```

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:

(Continúa)



(Continuación)

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 tres intentos de acceso fallidos para la cuenta, el cuarto producirá el bloqueo de la cuenta:

```
CREATE PROFILE PERFILE2 LIMIT FAILED_LOGIN_ATTEMPTS 3;
```

Se asigna el perfil al usuario PRUEBA:

```
ALTER USER PRUEBA PROFILE PERFILE2;
```

Si hay tres 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;
```

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 PERFILE2 CASCADE;
```



9. Administración de Oracle

9.5 Gestión de tablespaces

9.5 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**, la cual 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 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];
```

Donde:

- 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.

9. Administración de Oracle

9.5 Gestión de tablespaces



- **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.
- **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}}
```

- **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.

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.

En Oracle 10g el tamaño, por defecto, de un bloque de datos es de 8k.



9. Administración de Oracle

9.5 Gestión de tablespaces



Caso práctico

- 7 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* 'TRABAJO01.ORA', de 10 M, y 'TRABAJO02.ORA', de 5 M:

```
CREATE TABLESPACE TRABAJO DATAFILE 'TRABAJO01.ORA' SIZE 10M,  
  'TRABAJO02.ORA' SIZE 5M  
  DEFAULT STORAGE (INITIAL 10K NEXT 10K PCTINCREASE 25);
```

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

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

• 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 definido en bytes (BYTES), el máximo tamaño que puede llegar a tener (MAXBYTES), etcétera. 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: Nº. de identificación del archivo.

Donde: BYTES: Es el Nº. de bytes libres.

BLOCK_ID: Identificación del primer bloque libre.

BLOCKS: Nº. 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.
- DBA_TABLESPACES: Muestra la descripción de todos los tablespaces.

9. Administración de Oracle

9.5 Gestión de tablespaces



- **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: N.º de bytes usados por el usuario.

MAX_BYTES: N.º máximo de bytes que tiene el usuario asignados. Si el valor es -1 significa que tiene asignado un n.º de bytes ilimitado.

BLOCKS: N.º de bloques usados.

MAX_BLOCKS: Máximo N.º de bloques.

Actividades propuestas

- 4 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* y 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:

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

Donde:

- *nombretablespace* es el nombre del *tablespace* que se quiere modificar.
- **ADD_DATAFILE** añade al *tablespace* uno o varios archivos.



9. Administración de Oracle

9.5 Gestión de tablespaces

- 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).

A continuación, se crea un *tablespace* TRABAJO con el tamaño base de 100 MB y cada extensión subsiguiente será de 100 MB.

Algunas veces incluye el comando:

CREATE TABLESPACE TRABAJO



Caso práctico

8 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

5 Se trata de 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 Megabyte.

Sigue insertando filas hasta que sobrepase 1 Megabyte 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 200 K sin indicar máximo tamaño. Prueba de nuevo insertar filas.

9. Administración de Oracle

9.5 Gestión de tablespaces



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.

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;
```

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 distintos a los que asigna el *tablespace*. El formato **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];
```

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.

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



9. Administración de Oracle

9.6 Otros objetos

9.6 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

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

Una **secuencia** es un objeto de la 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 1027 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 -1026 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.

9. Administración de Oracle

9.6 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: NOMBRESECUENCIA.CURRVAL NOMBRESECUENCIA.NEXTVAL.

Caso práctico

- 9 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;

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

```
DROP SEQUENCE CODIGOS;
```

Actividades propuestas

- 6 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.



9. Administración de Oracle

9.6 Otros objetos

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 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 y, además, cada fila se identifique por 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 crea 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;`



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 nombreddeenlace
  CONNECT TO usuario IDENTIFIED BY clave
  USING 'cadena_de_conexion';
```

Donde:

- PUBLIC crea un enlace de base de datos público.
- *cadena_de_conexion* 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

- ⑩ 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 nombreddeenlace;
```

Por ejemplo, borramos el enlace anterior:

```
DROP DATABASE LINK MIENLACE;
```



9. Administración de Oracle

Conceptos básicos

8. Índices

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 un *tablespace* de nombre **COMPRAS** asociándole un fichero en disco llamado '**COMPRAS.ORA**' de 5 Megabytes.

2 Modifica el *tablespace* del Ejercicio 1 para que pueda autoextenderse automáticamente, sin límite de espacio en disco.

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**, al cual se le asigna 1 Megabyte. El *tablespace* temporal será **TEMP**. Se le asigna el rol anterior.

Realiza la siguiente secuencia de instrucciones en el orden indicado:

- Crea un usuario de base de datos que tenga funciones de administrador.
- Conéctate con el nombre de usuario creado.
- Crea varias tablas en el propio esquema.
- Crea cinco usuarios nuevos asignándoles un *tablespace* por defecto y cuota (**USU1**, **USU2**, **USU3**, **USU4** y **USU5**).
- Da permiso a uno de los usuarios (**USU1**) para que sólo pueda conectarse a la base de datos.

f) Crea un rol que permita conectarse a la base de datos y hacer **SELECT** sobre algunas tablas.

g) Concede el rol creado a dos de los usuarios creados anteriormente (**USU2** y **USU3**).

h) Concede al usuario **USU4** privilegios sobre algunas tablas con la opción de poder concedérselos a otros usuarios.

i) Concede al usuario **USU5** cuatro privilegios de sistema, dos de ellos, con la opción de poder concedérselos a otros usuarios.

j) Concede a todos los usuarios de la base de datos privilegios para que puedan modificar ciertas columnas de algunas tablas.

k) Quita a los usuarios **USU3** y **USU4** todos los privilegios que tenían asignados.

l) Haz que **USU5** sólo pueda conectarse en dos sesiones concurrentes a la vez.

m) Limita el tiempo de conexión a la base de datos a cinco minutos a los usuarios **USU2** y **USU3**.

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 utilizado en alguna de las columnas.