



TEMA



## Tema 2. Tablespaces y privilegios en Oracle

Administración de sistemas  
informáticos en red

**Administración de sistemas gestores  
de bases de datos**

Autor: Joan Pou



## Tema 2: Tablespaces y privilegios en Oracle

### ¿Qué aprenderás?

---

- Instalar Oracle.
- Gestionar los espacios de almacenamiento.
- Utilizar SQL PLUS y SQL developer.
- Crear conexiones a la base de datos.
- Crear y gestionar usuarios.
- Asignar permisos y privilegios.
- Crear y gestionar roles.
- Crear y gestionar perfiles.

### ¿Sabías que...?

---

- Oracle es una de las mayores compañías de software del mundo.
- Actualmente dispone de más de 420.000 clientes.



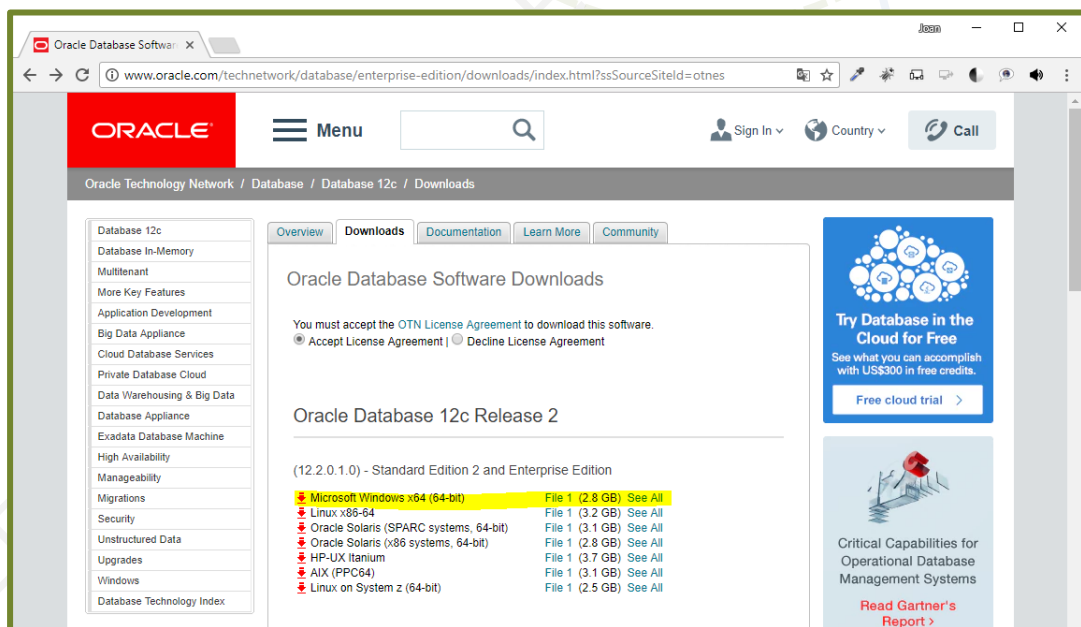
## 2. Tablespaces y privilegios en Oracle

### 2.1. Instalación de Oracle

Oracle la Primera Base de Datos Diseñada para Grid Computing, es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general.

Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer. Actualmente dispone de más de 420 000 clientes e implementaciones en más de 145 países.

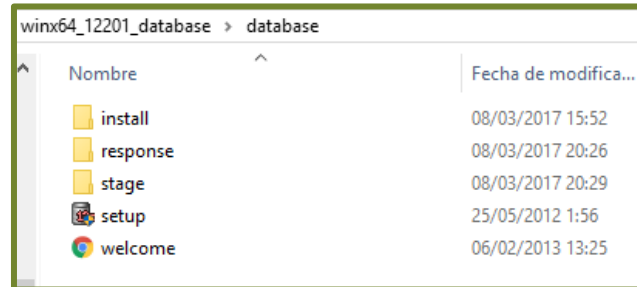
Accedemos a la página web de Oracle y descargamos el fichero correspondiente a la última versión.



<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html?ssSourceSitelId=otnes>



- Deberemos aceptar los términos de la licencia y registrarnos en Oracle para poder descargar el software.
- Descomprimos el fichero en una carpeta y ejecutamos el setup.



Entramos en un asistente que nos va a guiar durante el proceso de instalación:

- Paso 1: Facilitamos una cuenta de correo electrónico para que nos manden información sobre seguridad y noticias de la base de datos.
- Paso 2: Seleccionamos: Crear y configurar la base de datos.
- Paso 3: Clase Escritorio, utiliza menos recursos y configuración que la versión servidor.
- Paso 4: Utilizamos la cuenta incorporada en Windows para ejecutar Oracle.
- Paso 5: Especificamos los directorios de trabajo, la edición de la base de datos (Standard o Enterprise) el juego de caracteres y desmarcamos Crear como Base de Datos de contenedor.



Con la versión Enterprise podemos realizar particionado de base de datos, paralelismo (utilizar varias CPU's) cores de forma simultánea y la posibilidad de utilizar los packs de Diagnostics y de Tuning, entre otras.

- Paso 6: Comprueba los requisitos del sistema. Muchas veces da errores de Path que podemos ignorar.
- Paso 7: Resumen de todas las opciones seleccionadas.
- Paso 8: Instalación del producto. En función de las características del ordenador puede tardar entre 10 y 20 minutos.

Una vez finalizado el proceso podemos comprobar que funciona correctamente abriendo una sesión en la aplicación de escritorio SQL PLUS.



```
SQL*Plus

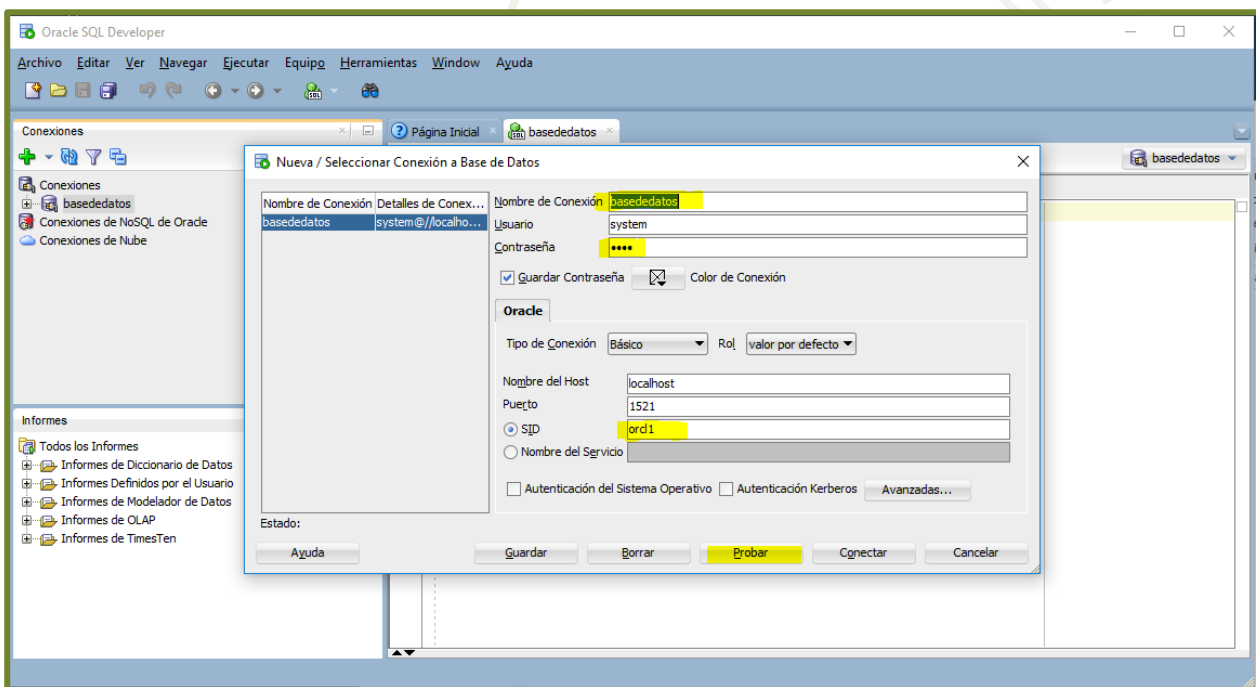
SQL*Plus: Release 12.2.0.1.0 Production on Sáb Sep 30 17:22:11 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Introduzca el nombre de usuario: system
Introduzca la contraseña:
Hora de última Conexión Correcta: Sáb Sep 30 2017 17:11:28 +02:00

Conectado a:
Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit Production
```

También podemos utilizar la herramienta gráfica SQL Developer, en este caso deberemos configurar una conexión a la base de datos.



Podemos crear una conexión llamada basededatos el nombre de usuario SYSTEM con la contraseña especificada durante el proceso de instalación. El SID de la base de datos lo hemos especificado también durante la instalación normalmente es ORCL. Podemos probar la conexión antes de utilizarla.

Para no trabajar directamente con el usuario SYSTEM podemos crear un nuevo usuario y darle permisos de administrador con las siguientes instrucciones:

```
CREATE USER usuario IDENTIFIED BY "password";  
GRANT DBA TO usuario;
```



## 2.2. Tablespaces en Oracle

Un tablespace es una unidad lógica de almacenamiento dentro de una base de datos de Oracle. Son estructuras donde se almacenan los objetos del esquema de la base de datos: tablas, vistas, índices, etc. Antes de crear las tablas en la base de datos se deberá crear un tablespace.

Los tablespaces se componen como mínimo de un datafile, y cada datafile solo puede pertenecer a un tablespace. Existen diferentes tipos de tablespaces, entre los que podemos destacar:

- **SYSTEM:** Contiene el diccionario de datos. Se crea automáticamente al hacer la instalación de Oracle.
- **SYSAUX:** Es un tablespace auxiliar donde se almacenan algunos de los componentes que antes de almacenaban en el tablespace SYSTEM.
- **TEMP:** Contiene los objetos temporales, una de sus utilidades es para optimizar las operaciones de ordenación.
- **BIGFILE:** Es un tipo especial de tablespace que almacena un solo datafile muy grande de hasta 128 TB.
- **USERS:** Usado para almacenar permanentemente objetos y datos de usuarios.
- **UNDOTBS1:** Se utiliza para almacenar información de Undo (deshacer).
- **EXAMPLE:** Contiene ejemplos de la base de datos que pueden ser instalados.

Para crear un tablespace utilizamos la orden **CREATE TABLESPACE**, su sintaxis completa es la siguiente:

```
CREATE [TEMPORARY / UNDO] TABLESPACE nombre_del_tablespace  
DATAFILE opciones_datafile opciones_almacenamiento;
```

Las opciones de datafile se pueden resumir en:

'ruta y nombre del fichero' SIZE entero [K|M] [AUTOEXTEND OFF]

'ruta y nombre del fichero' SIZE entero [K|M] [AUTOEXTEND ON [NEXT int K|M] [MAXSIZE int K|M] ]

- **SIZE:** especifica el tamaño del tablespace en Kilobytes(K) o en Megabytes(M).
- **AUTOEXTEND:** activa o desactiva el crecimiento automático del tablespace. NEXT especifica el incremento y MAXSIZE el tamaño máximo de espacio reservado en el disco. MAXSIZE toma el valor por defecto UNLIMITED si no se especifica ningún valor.



Las opciones de almacenamiento se pueden resumir en:

**DEFAULT STORAGE (**

INITIAL entero	el tamaño de la primera extensión
NEXT entero	el tamaño de la segunda extensión
MINEXTENTS entero	número inicial de extensiones
MAXEXTENT entero	número máximo de extensiones
PCTINCREASE entero)	% incremento de tamaño en las siguientes extensiones

**ONLINE | OFFLINE**

**PERMANENT | TEMPORARY**

**EXTENT MANAGEMENT {DICTIONARY |**

**LOCAL {AUTOALLOCATE | UNIFORM [SIZE entero K | M]} }**

Cuando creamos un tablespace podemos definir un almacenamiento por defecto para los objetos que se crean dentro de él. Debemos tener en cuenta que una extensión en Oracle es un conjunto de bloques de datos contiguos (unidad mínima que escribe en el disco).

- **OFFLINE|ONLINE.** Desactivamos o activamos el acceso al tablespace, por defecto **ONLINE**.
- **PERMANENT | TEMPORARY.** Si indicamos la opción **TEMPORARY** el tablespace solo podrá almacenar objetos temporales, por defecto **PERMANENT**.
- **EXTENT MANAGEMENT** especifica como gestionamos las extensiones del tablespace. **DICTIONARY:** mediante tablas del diccionario, **LOCAL:** mediante un mapa de bits, **AUTOALLOCATE:** el tablespace lo gestiona el sistema y **UNIFORM:** se gestiona con extensiones de un tamaño determinado.

## 2.3. Modificar un tablespace

Las modificaciones que se pueden aplicar a un tablespace una vez creado son generalmente las de activar o desactivar, añadir nuevos archivos, modificar los existentes, etc. Utilizamos la orden **ALTER TABLESPACE** con la siguiente sintaxis:

```
ALTER TABLESPACE nombre_del_tablespace  
[ADD DATAFILE 'nombre' SIZE entero [K|M] [AUTOEXTEND [OFF|ON]]  
[RENAME DATAFILE 'archivo' TO 'archivo]  
[ONLINE|OFFLINE]
```

- **ADD DATAFILE:** para añadir uno o mas archivos al tablespace.
- **RENAME DATAFILE:** Cambia el nombre del fichero, primero debe realizarse desde el sistema operativo.
- **ONLINE:** activa el tablespace y **OFFLINE:** desactiva el tablespace.





## 2.4. Borrar un tablespace

Para borrar un tablespace utilizamos la orden DROP TABLESPACE, su sintaxis es la siguiente:

```
DROP TABLESPACE nombre_del_tablespace  
[INCLUDING CONTENTS [AND DATAFILES]];
```

La opción INCLUDING CONTENTS nos permite borrar un tablespace que contenga datos y AND DATAFILES elimina los archivos de datos asociados a este tablespace.

## 2.5. Consultar tablespaces

Existen un conjunto de vistas con las que podemos obtener información sobre los tablespaces. Algunas de ellas son:

- **DBA\_TABLESPACES:** Muestra información relativa a los tablespaces, con ella podemos obtener por ejemplo el nombre(TABLESPACE\_NAME) y el tipo de tablespace(CONTENTS).
- **DBA\_DATA\_FILES:** Nos muestra información sobre los archivos que utiliza el TABLESPACE.
- **DBA\_TS\_QUOTAS:** Muestra en cada tablespace los bytes que ha utilizado cada usuario.
- **USER\_FREE\_SPACE:** Muestra las extensiones libres en los tablespaces a los que puede acceder el usuario.

Ejemplos:

- Creamos un tablespace llamado pruebas almacenado en un fichero 'pruebas.dbf' con un tamaño de 100MB

```
create tablespace pruebas datafile 'c:\oracle\pruebas.dbf' size 100M autoextend on;
```

- Comprobamos el tamaño del tablespace.

```
select tablespace_name, bytes from dba_data_files;
```



```
SQL> create tablespace pruebas datafile 'c:\oracle\pruebas.dbf'
      2 size 100M autoextend on;

Tablespace creado.

SQL> select tablespace_name, bytes from dba_data_files;

TABLESPACE_NAME          BYTES
-----
SYSTEM                   838860800
SYSAUX                   734003200
UNDOTBS1                  807403520
USERS                     5242880
EXAMPLE                  1319895040
PRUEBAS                   104857600

6 filas seleccionadas.
```

- Listamos el nombre y el tipo de todos los tablespaces.

**select tablespace\_name, contents from dba\_tablespaces;**

```
SQL> select tablespace_name, contents from dba_tablespaces;

TABLESPACE_NAME          CONTENTS
-----
SYSTEM                   PERMANENT
SYSAUX                   PERMANENT
UNDOTBS1                  UNDO
TEMP                     TEMPORARY
USERS                     PERMANENT
EXAMPLE                  PERMANENT
PRUEBAS                   PERMANENT

7 filas seleccionadas.
```

- Borramos el tablespace pruebas y los datafiles.

**drop tablespace pruebas including contents and datafiles;**

- Creamos el tablespace pruebas con 100M.

**create tablespace pruebas datafile 'c:\oracle\pruebas.dbf' size 100M autoextend on;**

- Modificamos el tablespace y lo redimensionamos a 150MB. Cambiando el tamaño del fichero.

**alter database datafile 'c:\oracle\pruebas.dbf' resize 150M;**



- Modificamos el tablespace y le añadimos un segundo fichero de 50MB llamado pruebas2.

**alter tablespace pruebas add datafile 'c:\oracle\pruebas2.dbf' size 50M;**

- Comprobamos el tamaño del tablespace.

**select tablespace\_name, bytes from dba\_data\_files;**

```
SQL> alter database datafile 'c:\oracle\pruebas.dbf' resize 150M;
Base de datos modificada.

SQL> alter tablespace pruebas
  2  add datafile 'c:\oracle\pruebas2.dbf' size 50M;
Tablespace modificado.

SQL> select tablespace_name, bytes from dba_data_files;

TABLESPACE_NAME          BYTES
-----
SYSTEM                   838860800
SYSAUX                   734003200
UNDOTBS1                 807403520
USERS                    5242880
EXAMPLE                 1319895040
PRUEBAS                 157286400
PRUEBAS                 52428800

7 filas seleccionadas.
```

## 2.6. Usuarios

Una cuenta de usuario incluye la información que necesita un usuario para iniciar la sesión en el sistema gestor de bases de datos y poder acceder a determinados objetos en función de los privilegios que concede el administrador.

Durante el proceso de instalación de Oracle se crean dos cuentas administrativas y otras dos con permisos especiales para tareas de optimización y monitorización de la base de datos.

- SYS. Tiene el rol de DBA y en su esquema se crea el diccionario de datos, donde se almacena información sobre el resto de las estructuras de la base de datos, por lo que no conviene de ninguna manera crear otro tipo de elementos en su esquema.
- SYSTEM. Posee también el rol DBA y se crea durante la instalación. La contraseña MANAGER que tiene por defecto se debería cambiar en la instalación. En su esquema se suelen crear tablas y vistas administrativas (pero no se deberían crear otro tipo de tablas). Para crear a otros usuarios es necesario conectarse como usuario SYSTEM.
- SYSMAN. Usado para realizar tareas administrativas con la aplicación Enterprise Manager.
- DBSMNP. Monitoriza Enterprise Manager.



### 2.6.1. Creación de usuarios

Para crear un usuario en Oracle debes disponer del privilegio CREATE USER, para crear usuarios por primera vez deberemos conectarnos con el usuario SYSTEM y la contraseña especificada en la instalación. Utilizaremos la orden CREATE USER con la siguiente sintaxis:

```
CREATE USER nombre IDENTIFIED BY contraseña  
[DEFAULT TABLESPACE nombre_del_tablespace]  
[TEMPORARY TABLESPACE tablespace_temporal]  
[QUOTA [entero K|M |UNLIMITED] ON nombre_del_tablespace]  
[PROFILE perfil]  
[PASSWORD EXPIRE]  
[ACCOUNT LOCK|UNLOCK];
```

- El nombre del usuario no puede repetirse y como máximo debe tener 30 caracteres que sólo podrán contener letras del alfabeto inglés, números, el signo del dólar o el signo del guión bajo.
- DEFAULT y TEMPORARY TABLESPACE asignan un tablespace de trabajo y temporal al usuario. Si no especifica el tablespace DEFAULT se asigna por defecto el de USERS.
- QUOTA asignar el límite de almacenamiento en KB o MB o bien ilimitado.
- PROFILE asigna un perfil al usuario. Los perfiles sirven para asignar el tiempo de CPU, el tiempo de la sesión, el número de conexiones concurrentes.
- PASSWORD EXPIRE. Especifica que la contraseña asignada al usuario caducará, de esta forma, el propio usuario o el DBA deberá asignar una de nueva antes de acceder al sistema.
- ACCOUNT LOCK. Bloquea la cuenta del usuario en el momento de la creación. Por defecto se crea desbloqueada.

La vista DBA\_USERS muestra la lista y configuración de todos los usuarios del sistema. Para observar la estructura de la vista, siempre es conveniente usar DESCRIBE DBA\_USERS.



## 2.6.2. Modificación y borrado de usuarios

Para modificar un usuario en Oracle utilizamos la orden ALTER USER y el nombre del usuario, con los mismos parámetros que la instrucción CREATE USER. Podemos cambiar la contraseña, los tablespaces, el perfil o la cuota.

Para borrar un usuario utilizamos la orden DROP USER con la siguiente sintaxis:

**DROP USER nombre\_usuario [CASCADE];**

- CASCADE. Suprime todos los elementos del usuario antes de borrarlo. Si intentamos borrar un usuario sin esta opción y tiene elementos creados nos dará un mensaje de error.

Ejemplos:

- Creamos el tablespace tienda almacenado en un fichero llamado tienda con 100MB.

**create tablespace tienda datafile 'c:\oracle\tienda.dbf' size 100M;**

- Creamos un usuario llamado usuario1 con contraseña user1. Lo asignamos al tablespace sin límite de tamaño.

**create user usuario1 identified by "user1" default tablespace tienda  
quota unlimited on tienda;**

- Creamos un usuario llamado usuario2 con contraseña user2. Lo asignamos al tablespace con un límite de 15M y le bloqueamos la cuenta.

**create user usuario2 identified by "user2" default tablespace tienda  
quota 15M on tienda account lock;**



```
SQL> create tablespace tienda datafile 'c:\oracle\tienda.dbf'
  2 size 100M;

Tablespace creado.

SQL> create user usuario1 identified by "user1"
  2 default tablespace tienda
  3 quota unlimited on tienda;

Usuario creado.

SQL> create user usuario2 identified by "user2"
  2 default tablespace tienda
  3 quota 15M on tienda account lock;

Usuario creado.
```

- Desbloqueamos al usuario2.  
**alter user usuario2 account unlock;**
- Asignamos espacio ilimitado al usuario2.  
**alter user usuario2 quota unlimited on tienda;**

```
SQL> alter user usuario2 account unlock;

Usuario modificado.

SQL> alter user usuario2 quota unlimited on tienda;

Usuario modificado.

SQL> drop user usuario1;

Usuario borrado.

SQL> select user_id, substr(username,1,30) from dba_users;

  USER_ID SUBSTR(USERNAME,1,30)
-----
    144 USUARIO2
    142 USER3
    139 USER1
    140 USER2
    133 JOAN
    130 MARTA
    109 OE
    110 SCOTT
     36 ORACLE_OCM
     70 OJVMSYS
```

- Borramos al usuario1.  
**drop user usuario1;**



- Listamos todos los usuarios del sistema.

```
select user_id, substr(username,1,30) from dba_users;
```

## 2.7. Privilegios

Los privilegios son permisos que damos a los usuarios para que puedan realizar ciertas operaciones con la base de datos. Los administradores son los encargados de crear los roles y asignarlos a los usuarios. En Oracle existen cerca de 200 privilegios diferentes y se clasifican en: privilegios del sistema, privilegios de objeto y roles de usuario.

### 2.7.1. Privilegios del sistema

Permiten al usuario hacer ciertas tareas sobre la base de datos, como por ejemplo crear una tabla, una vista o un tablespace. Estos privilegios los otorga el administrador o algún otro usuario que haya recibido el permiso para administrar estos privilegios.

Los privilegios de sistema permiten ejecutar comandos del tipo DDL o del tipo DML. Existen también dos privilegios especiales que permiten conceder nivel de DBA y son: SYSDBA y SYSOPER.

Entre los privilegios de sistema podemos destacar los siguientes:

Objeto	Privilegio del sistema	Operación autorizada
TABLE	CREATE TABLE	Crear tablas en nuestro esquema.
	CREATE ANY TABLE	Crear tablas en cualquier esquema.
	ALTER ANY TABLE	Modificar tablas en cualquier esquema.
	DROP ANY TABLE	Borrar tablas en cualquier esquema.
	LOCK ANY TABLE	Bloquear tablas 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 en cualquier tabla.



TABLESPACE	CREATE TABLESPACE	Crear tablespaces.
	ALTER TABLESPACE	Modificar tablespaces.
	MANAGE TABLESPACE	Poner online u offline cualquier tablespace.
	DROP TABLESPACE	Borrar un tablespace.

Objeto	Privilegio del sistema	Operación autorizada
USER	CREATE USER	Crear usuarios y cuotas de tablespaces.
	ALTER USER	Modificar usuarios: contraseña, cuota, etc.
	DROP USER	Borrar 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.
SESSION	CREATE SESSION	Conectarnos a la base de datos.
ROLE	CREATE ROLE	Crear roles.
	ALTER ANY ROLE	Modificar roles.
	DROPE ANY ROLE	Borrar cualquier rol.
	GRANT ANY ROLE	Dar permisos para cualquier rol de la base de datos.
PROCEDURE	CREATE PROCEDURE	Crear procedimientos y funciones.
	CREATE ANY PROCEDURE	Crear procedimientos y funciones en cualquier esquema.
	ALTER ANY PROCEDURE	Modificar proced. y funciones de cualquier esquema.
	DROP ANY PROCEDURE	Borrar procedimientos y funciones de cualquier esquema
	EXECUTE ANY PROCEDURE	Ejecutar procedimientos y funciones.
PRIVILEGE	GRANT ANY PRIVILEGE	Conceder cualquier privilegio de sistema.





La tabla anterior recoge un resumen de los privilegios más utilizados, pero para otros objetos el funcionamiento es similar: igual que hay un CREATE TABLE, se puede utilizar un CREATE INDEX, TRIGGER, SEQUENCE, SYNONYM, TYPE,... y de esa forma podemos conceder el privilegio de creación a otros objetos. Lo mismo con el resto de operaciones.

Los privilegios a nivel de administración son SYSDBA y SYSOPER. Las operaciones autorizadas son las siguientes:

Privilegio	Operación autorizada.
SYSDBA	<ul style="list-style-type: none"><li>• Iniciar y parar el servidor de base de datos (STARTUP y SHUTDOWN).</li><li>• ALTER DATABASE: open, mount, backup, o change character set.</li><li>• CREATE DATABASE</li><li>• DROP DATABASE</li><li>• CREATE SPFILE</li><li>• ALTER DATABASE ARCHIVELOG</li><li>• ALTER DATABASE RECOVER</li><li>• Incluye el privilegio RESTRICTED SESSION</li></ul>
SYSOPER	<ul style="list-style-type: none"><li>• Iniciar y parar el servidor de base de datos (STARTUP y SHUTDOWN).</li><li>• CREATE SPFILE</li><li>• ALTER DATABASE ARCHIVELOG</li><li>• ALTER DATABASE RECOVER</li><li>• Incluye el privilegio RESTRICTED SESSION</li></ul>

El privilegio de SYSOPER a diferencia del SYSDBA permite a un usuario realizar las tareas básicas de funcionamiento, pero sin la capacidad de mirar los datos del usuario.

Para asignar los privilegios de objeto utilizamos la orden GRANT con la siguiente sintaxis:

```
GRANT permiso1 [,permiso2, ...]  
TO nombreUsuario [,nombreusuario2, ...] | nombreRol  
[WITH ADMIN OPTION];
```



- TO: Usuario o rol al que se conceden los privilegios.
- WITH ADMIN OPTION. El que recibe el privilegio puede concederlo a otros usuarios o roles.

Ejemplos:

- Permiso de iniciar sesión al usuario1 y para crear tablas.

**grant create session, create table to usuario1;**

- Concedemos el permiso de crear y ejecutar procedimientos almacenados al usuario2.

**grant create procedure, execute any procedure to usuario2;**

- Concedemos el permiso de crear usuarios al usuario1 y este usuario los podrá conceder a otros usuarios.

**grant create user to usuario1 with admin option;**

- Concedemos permisos de DBA al usuario3.

**grant dba to usuario3;**

Para quitar privilegios sobre el sistema utilizamos la orden REVOKE con la siguiente sintaxis:

**REVOKE permiso1[,permiso2,...] | ALL [PRIVILEGES]  
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];**

- FROM: Usuario al que se quitan los privilegios.
- ALL: Permite quitar todos los permisos.
- PUBLIC: Quita el privilegio o privilegios a todos los usuarios del sistema.

Ejemplos:

- Quitamos todos los privilegios al usuario1.

**revoke all privileges from usuario1;**

- Quitamos el privilegio de crear tablas al usuario2.

**revoke create table from usuario2;**



### 2.7.2. Privilegios de objeto

Permiten al usuario realizar ciertas acciones en objetos de la base de datos, como una tabla, vista, procedimiento, etc. Si al usuario no se le dan estos permisos, sólo puede acceder a sus propios objetos. Estos privilegios los puede conceder: el administrador, el propietario del objeto u otro usuario que haya recibido el permiso con la opción GRANT OPTION. Por ejemplo consultar la tabla de otro usuario es un privilegio sobre objetos.

Algunos de los privilegios de objeto más usados son:

Permiso	Tabla	Vista	Secuencia	Procedimiento
ALTER	X		X	
UPDATE	X			
DELETE	X	X		
EXECUTE				X
INSERT	X	X		
SELECT	X	X	X	

Para asignar los privilegios de objeto utilizamos la orden GRANT con la siguiente sintaxis:

```
GRANT privilegio1 [[,privilegio2, ...] | ALL]  
[(columna1[,columna2,...])]  
[ON usuario[.objeto] | ANY TABLE]  
TO {nombreUsuario | rol | PUBLIC}  
[WITH GRANT OPTION];
```

- ON: Objeto sobre el que se aplican los privilegios.
- TO: Usuario al que se conceden los privilegios.
- ALL: Permite asignar todos los permisos.
- PUBLIC: Asigna el privilegio o privilegios a todos los usuarios del sistema.
- WITH GRANT OPTION: Permite que el usuario que lo reciba pueda conceder permisos a otros usuarios.



Ejemplos:

- Permiso de modificación en cualquier tabla al usuario2.

**grant update any table to usuario2;**

- Concedemos el permiso de select sobre la tabla productos del usuario1 al usuario2 y le permitimos que lo conceda a otros usuarios.

**grant select on usuario1.productos to usuario2 with grant option;**

- Concedemos los permisos de seleccionar, insertar y modificar a todos los usuarios sobre la tabla productos del usuario1.

**grant select, insert, update on usuario1.productos TO public;**

- Concedemos el permiso de modificar sólo el nombre y el precio de la tabla productos al usuario3.

**GRANT UPDATE (nombre, precio) ON usuario1.productos TO usuario3;**

Para quitar privilegios sobre objetos utilizamos la orden REVOKE con la siguiente sintaxis:

**REVOKE permiso1[,permiso2,...] | ALL [PRIVILEGES]  
ON [usuario.]objeto  
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];**

- ON: Objeto sobre el que se quitan los privilegios.
- FROM: Usuario al que se quitan los privilegios.
- ALL: Permite quitar todos los permisos.
- PUBLIC: Quita el privilegio o privilegios a todos los usuarios del sistema.

Ejemplos:

- Quitamos el permiso de selección al usuario2 sobre la tabla productos del usuario1.

**REVOKE SELECT ON usuario1.productos FROM usuario2;**

- Quitamos el privilegio de modificación a todos los usuarios sobre cualquier tabla.

**revoke update any table from public;**



### 2.7.3. Mostrar información sobre los privilegios

Disponemos de las siguientes vistas que nos dan información sobre los privilegios concedidos a los diferentes objetos del sistema.

- DBA\_SYS\_PRIVS. Privilegios de sistema asignados a usuarios y roles.
- DBA\_TAB\_PRIVS. Lista de todos los privilegios de todos los objetos de la BD.
- DBA\_COL\_PRIVS. Lista de todos los privilegios aplicados a columnas de la BD.
- SESSION\_PRIVS. Privilegios en activo para el usuario y sesión actuales.

Ejemplo:

- Concedemos el permiso de crear sesión al usuario2 y lo comprobamos.

```
SQL> grant create session to usuario2;
Concesión terminada correctamente.

SQL> select * from dba_sys_privs where grantee='USUARIO2';

GRANTEE  PRIVILEGE                                ADM COM
-----
USUARIO2 CREATE SESSION                  NO  NO

SQL>
```



[Video Usuarios](#)



## 2.8. Roles

Los roles son un conjunto de privilegios que se pueden otorgar a un usuario o a otro Rol. De esta manera se simplifica el trabajo de DBA. Los privilegios de un rol pueden ser a nivel de objeto o a nivel de sistema.

En una base de datos de Oracle existen muchos roles predefinidos. Los mas importantes y usados son los siguientes:

- **CONNECT:** Existe por compatibilidad con versiones anteriores en las que se incluían privilegios para crear objetos como tablas. Ahora solo incluye el privilegio CREATE SESSION.
- **RESOURCE:** Permite crear tablas y código PL/SQL del tipo que sea. Se mantiene por compatibilidad.
- **DBA:** Tiene la mayoría de los privilegios de sistema, sobre objetos y roles. Un usuario con el rol DBA puede administrar la base de datos, excepto el arranque y parada de la misma.

### 2.8.1. Creación de roles

Para crear un rol utilizamos la orden CREATE ROL con la siguiente sintaxis:

```
CREATE ROLE nombrol  
[NOT IDENTIFIED | IDENTIFIED BY password];
```

- **IDENTIFIED:** Sirve para asignar un password al ROL por defecto no requiere identificación.

Ejemplos:

- Crear un rol llamado rolprofesor.

```
CREATE ROLE rolprofesor;
```

- Crear un rol llamado administrador con contraseña admin1234.

```
CREATE ROLE administrador IDENTIFIED BY "admin1234";
```

Para asignar y retirar privilegios a los roles utilizamos la orden GRANT vista anteriormente. A los roles se les asignan privilegios igual que a los usuarios, pueden ser de sistema y/ o de objeto. Para eliminar los privilegios de un rol utilizamos REVOKE.



Ejemplos:

- Asignar el privilegio de create session al rol profesor.

**GRANT CREATE SESSION TO rolprofesor;**

- Asignar el privilegio de select, insert y update al rol profesor sobre la tabla alumnos.

**GRANT SELECT, INSERT, UPDATE ON alumnos TO rolprofesor;**

Para quitar privilegios a un rol utilizamos la orden REVOKE con la misma sintaxis que habíamos visto anteriormente.

Ejemplo:

- Quitar el privilegio de update al rol profesor sobre la tabla alumnos.

**REVOKE UPDATE ON alumnos FROM rolprofesor;**

### 2.8.2. Asignar roles a usuarios

Para asignar un rol a un usuario o a otro rol utilizamos la orden GRANT con la siguiente sintaxis:

**GRANT rol1 [,rol2, ...]  
TO nombreUsuario [,nombreusuario2, ...] | nombreRol | PUBLIC  
[WITH ADMIN OPTION];**

- TO: Usuario o rol al que se asigna el rol
- WITH ADMIN OPTION. El que recibe el privilegio puede conceder dicho rol a otros usuarios.



Ejemplos:

- Creamos dos usuarios llamados profe1 y profe2 con contraseña "p1234" y les asignamos el rol de profesor.

```
CREATE USER profe1 IDENTIFIED BY "p1234";  
CREATE USER profe2 IDENTIFIED BY "p1234";  
GRANT rolprofesor TO profe1, profe2;
```

- Asignamos el rol predefinido DBA al usuario administrador.

```
GRANT DBA TO administrador;
```

Para eliminar un rol de la base de datos utilizamos la orden DROP ROLE. Con esta orden retiramos el rol a todos los usuarios que lo poseen. Para poder eliminar un rol debemos tener el privilegio de DROP ANY ROLE o ser administrador.

Ejemplo:

- Eliminar el rol rolprofesor.

```
DROP ROLE rolprofesor;
```

También podemos establecer un rol por defecto a un determinado usuario utilizando la orden ALTER USER. Para poder aplicar esta instrucción el usuario debe tener concedido este ROL.

```
ALTER USER usuario DEFAULT ROLE nombrerol;
```

Ejemplo:

- El rol por defecto del usuario profe2 será rolprofesor.

```
ALTER USER profe2 DEFAULT ROLE rolprofesor;
```





### 2.8.3. Consulta de información sobre los roles

La base de datos dispone de una serie de vistas para saber a qué usuarios se ha concedido un rol, o los privilegios otorgados a un rol. Las vistas son:

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

Ejemplos:

- Para ver los roles definidos en la base de datos.

```
select role from dba_roles order by 1;
```

- Roles asignados a un usuario en concreto.

```
select grantee, granted_role from dba_role_privs where grantee= 'PROFE1';
```

- Roles del usuario conectado.

```
select username, granted_role from user_role_privs;
```



## 2.9. Perfiles

Un perfil de usuario sirve para limitar los recursos que puede utilizar un usuario en el sistema. Cada usuario puede tener un único perfil. Los recursos que podemos limitar son los recursos del kernel: uso de la CPU, duración de la sesión y también los límites de uso de los passwords: duración, intentos, etc.

Si a un usuario no se le asigna ningún perfil se utiliza el perfil por defecto (DEFAULT) que especifica recursos ilimitados para todos los usuarios.

Para crear un perfil se utiliza la orden CREATE PROFILE con la siguiente sintaxis:

```
CREATE PROFILE nombreperfil LIMIT  
{opciones recursos | opciones contraseña }  
{Entero [K|M] | UNLIMITED | DEFAULT};
```

- UNLIMITED: significa que no hay límite sobre un recurso.
- DEFAULT: Toma como límite el del perfil DEFAULT.

Las opciones de recursos y contraseña son las siguientes:

Recurso	Función
CONNECT_TIME	Tiempo permitido de conexión por sesión en minutos.
CPU_PER_CALL	Máximo tiempo de CPU por llamada (análisis, ejecución o búsqueda) en centésimas de segundo.
CPU_PER_SESSION	Tiempo máximo de CPU por sesión en centésimas de segundo.
IDLE_TIME	Tiempo máximo permitido en minutos sin actividad al usuario antes de ser desconectado.
LOGICAL_READS_PER_CALL	Máximo número de bloques leídos por llamada.
LOGICAL_READS_PER_SESSION	Máximo número de bloques leídos por sesión.
PRIVATE_SGA	Cantidad de espacio privado que una sesión puede reservar en la SGA (para servidores compartidos).



COMPOSITE_LIMIT	Suma del máximo de los límites anteriores. Si es excedido, Oracle aborta la sesión y devuelve un error.
SESSIONS_PER_USER	Número máximo de sesiones concurrentes permitidas por usuario.
FAILED_LOGIN_ATTEMPTS	Número máximo de intentos fallidos de un usuario antes de bloquear la cuenta.
PASSWORD_GRACE_TIME	Duración en días del período de gracia durante el cual puede cambiarse la contraseña cuando ha caducado.
PASSWORD_LIFE_TIME	Número de días de validez de una contraseña.
PASSWORD_LOCK_TIME	Número de días que permanecerá bloqueado un usuario después de rebasar el límite de FAILED_LOGIN_ATTEMPTS.
PASSWORD_REUSE_MAX	Número de veces que debe cambiar una contraseña antes de poder reutilizarla.
PASSWORD_REUSE_TIME	Número de días que deben pasar antes de que se pueda reutilizar una contraseña.
PASSWORD_VERIFY_FUNCTION	En este parámetro se puede indicar un script para validar la contraseña.

Ejemplo:

- Crear un perfil llamado perfil1 que limite las sesiones concurrentes por usuario a 2, el tiempo de conexión a 10 minutos, el cambio de contraseña a 30 días y 3 intentos de inicio de sesión.

```
CREATE PROFILE perfil1 LIMIT  
SESSIONS_PER_USER 2  
CONNECT_TIME 10  
PASSWORD_LIFE_TIME 30  
FAILED_LOGIN_ATTEMPTS 3;
```

Para asignar un perfil a un usuario utilizamos la orden ALTER USER con la siguiente sintaxis:

```
ALTER USER usuario PROFILE nombreperfil;
```



Ejemplo:

- Asignar el perfil1 al usuario2.

**ALTER USER usuario2 PROFILE perfil1;**

Para modificar un perfil utilizamos la orden ALTER PROFILE con el mismo formato que la orden CREATE PROFILE.

Ejemplo:

- Modificamos el límite de cambio de contraseña del perfil1 a 15 días.

**ALTER PROFILE perfil1 LIMIT  
PASSWORD\_LIFE\_TIME 15;**

Para borrar un perfil utilizamos la orden DROP PROFILE con la siguiente sintaxis:

**DROP PROFILE nombreperfil [CASCADE];**

- CASCADE: Esta opción se debe incluir si el perfil lo tiene asignado algún usuario

Ejemplo:

- Borramos el perfil1.

**DROP PROFILE perfil1 CASCADE;**

La vista DBA\_PROFILES contiene información sobre los límites.

Ejemplo:

- Listar los límites del perfil1.

**SELECT resource\_name, substr(limit,1,20)  
FROM DBA\_PROFILES WHERE profile='PERFIL1';**



```
SQL> SELECT resource_name, substr(limit,1,20)
       2 FROM DBA_PROFILES WHERE profile='PERFIL1';
```

RESOURCE_NAME	SUBSTR(LIMIT,1,20)
COMPOSITE_LIMIT	DEFAULT
SESSIONS_PER_USER	2
CPU_PER_SESSION	DEFAULT
CPU_PER_CALL	DEFAULT
LOGICAL_READS_PER_SESSION	DEFAULT
LOGICAL_READS_PER_CALL	DEFAULT
IDLE_TIME	DEFAULT
CONNECT_TIME	10
PRIVATE_SGA	DEFAULT
FAILED_LOGIN_ATTEMPTS	3
PASSWORD_LIFE_TIME	15
PASSWORD_REUSE_TIME	DEFAULT
PASSWORD_REUSE_MAX	DEFAULT
PASSWORD_VERIFY_FUNCTION	DEFAULT
PASSWORD_LOCK_TIME	DEFAULT
PASSWORD_GRACE_TIME	DEFAULT

16 filas seleccionadas.



**Linkia FP**

Formación Profesional Oficial a Distancia



[Video Roles y Perfiles](#)



## Recursos y enlaces

---

- [Administrar usuarios](#)



- [Administrar roles](#)



## Conceptos clave

---

- **DCL:** Lenguaje que permite gestionar el acceso a los datos.
- **GRANT:** Permite conceder un permiso o un privilegio.
- **REVOKE:** Permite quitar un permiso o un privilegio.
- **Roles:** Sirven para agrupar permisos y privilegios.
- **Perfiles:** Permiten limitar el acceso a los recursos.



## Test de autoevaluación

Un tablespace de oracle se utiliza para:

- a) Gestionar la política de usuarios del sistema.
- b) Almacenar los objetos de la base de datos.
- c) Asignar privilegios al sistema.
- d) Almacenar las copias de seguridad automáticas.

Con la vista dba\_data\_files obtenemos información de:

- a) Los archivos temporales del sistema.
- b) Las tablas de los usuarios.
- c) Los archivos que utiliza el tablespace.
- d) El diccionario de datos.

Para borrar un usuario del sistema utilizamos la instrucción:

- a) delete user
- b) remove user
- c) revoke user
- d) drop user

Para crear usuarios en el sistema utilizamos el lenguaje:

- a) DCL
- b) DML
- c) DLL
- d) DDL

Para consultar todos los usuarios del sistema podemos utilizar:

- a) La tabla system\_users
- b) La tabla users.
- c) La vista dba\_users.
- d) La vista ora\_users.



Para que un usuario pueda iniciar sesión le concedemos el permiso:

- a) connect
- b) login
- c) create session
- d) init

¿Cuáles de los siguientes son privilegios de objeto?

- a) INSERT
- b) SYSOPER
- c) SYSDBA
- d) SELECT

¿Un rol es?

- a) Un tipo de usuario.
- b) Un administrador de la base de datos.
- c) Un conjunto de privilegios.
- d) Una vista de usuarios.

Para consultar los roles de un determinado usuario utilizamos la vista:

- a) session\_roles
- b) role\_tab\_privs
- c) user\_role\_privs
- d) dba\_role\_privs

Para que un usuario sea administrador le concedemos el rol:

- a) ADMIN
- b) DBA
- c) RESOURCE
- d) DBA\_ADMIN





Un perfil de usuario sirve:

- a) Asignar privilegios.
- b) Configurar tipos de roles.
- c) Limitar recursos del sistema.
- d) Administrar sesiones.

Para asignar un perfil a un usuario utilizamos:

- a) ALTER USER usuario PROFILE perfil;
- b) ALTER PROFILE perfil TO usuario.
- c) CREATE perfil TO usuario.
- d) ALTER PROFILE perfil TO USER usuario.

## Ponlo en práctica

---

### Actividad 1

---

Crea un tablespace de datos llamado “concesionario” con un espacio inicial de 200MB y lo almacenas en un fichero llamado conce.dbf.

Muestra una lista de todos los tablespaces junto con su tamaño.

### Actividad 2

---

Crea una tabla llamada vehiculos donde se almacene la matricula, la marca, el modelo, los kilómetros y el precio de vehículos de segunda mano. Utilizar el tipo de campo y la longitud que creáis más adecuados para cada uno de los campos. Introduce datos en la tabla.

Crea un usuario llamado “vendedor” con contraseña “Venta1234” y lo asignas al tablespace concesionario. Este usuario puede iniciar sesión.

El usuario vendedor puede añadir, modificar y consultar datos en la tabla vehiculos.

Comprueba que se han asignado correctamente los permisos.



### Actividad 3

---

Crea un rol ventas que pueda con los permisos de conectar, leer, modificar e insertar datos en la tabla de vehículos.

Crea un usuario llamado comercial1 con contraseña com1234 y le asignas el rol ventas. Comprueba que tiene los permisos correspondientes.

Crea un perfil llamado perfilventas que le obligue a cambiar la contraseña cada 30 días. Asigna el perfil al usuario comercial.



## SOLUCIONARIOS

### Test de autoevaluación

---

Un tablespace de oracle se utiliza para:

- e) Gestionar la política de usuarios del sistema.
- f) **Almacenar los objetos de la base de datos.**
- g) Asignar privilegios al sistema.
- h) Almacenar las copias de seguridad automáticas.

Con la vista dba\_data\_files obtenemos información de:

- e) Los archivos temporales del sistema.
- f) Las tablas de los usuarios.
- g) **Los archivos que utiliza el tablespace.**
- h) El diccionario de datos.

Para borrar un usuario del sistema utilizamos la instrucción:

- e) delete user
- f) remove user
- g) revoke user
- h) **drop user**

Para crear usuarios en el sistema utilizamos el lenguaje:

- e) **DCL**
- f) DML
- g) DLL
- h) DDL



Para consultar todos los usuarios del sistema podemos utilizar:

- e) La tabla system\_users
- f) La tabla users.
- g) **La vista dba\_users.**
- h) La vista ora\_users.

Para que un usuario pueda iniciar sesión le concedemos el permiso:

- e) connect
- f) login
- g) **create session**
- h) init

¿Cuáles de los siguientes son privilegios de objeto?

- e) **INSERT**
- f) SYSOPER
- g) SYSDBA
- h) **SELECT**

¿Un rol es?

- e) Un tipo de usuario.
- f) Un administrador de la base de datos.
- g) **Un conjunto de privilegios.**
- h) Una vista de usuarios.

Para consultar los roles de un determinado usuario utilizamos la vista:

- e) **session\_roles**
- f) role\_tab\_privs
- g) user\_role\_privs
- h) dba\_role\_privs



Para que un usuario sea administrador le concedemos el rol:

- e) ADMIN
- f) **DBA**
- g) RESOURCE
- h) DBA\_ADMIN

Un perfil de usuario sirve:

- e) Asignar privilegios.
- f) Configurar tipos de roles.
- g) **Limitar recursos del sistema.**
- h) Administrar sesiones.

Para asignar un perfil a un usuario utilizamos:

- e) **ALTER USER usuario PROFILE perfil;**
- f) ALTER PROFILE perfil TO usuario.
- g) CREATE perfil TO usuario.
- h) ALTER PROFILE perfil TO USER usuario.



## Ponlo en práctica

### Actividad 1

Crea un tablespace de datos llamado “concesionario” con un espacio inicial de 200MB y lo almacenas en un fichero llamado conce.dbf.

Muestra una lista de todos los tablespaces junto con su tamaño.

#### Solución:

```
create tablespace concesionario datafile 'E:\datos\conce.dbf' size 200M;
```

```
select tablespace_name, sum(bytes) total from dba_data_files  
group by tablespace_name;
```

	TABLESPACE_NAME	TOTAL
1	SYSAUX	702545920
2	UNDOTBS1	560988160
3	EMPRESA	104857600
4	USERS	5242880
5	CONCESIONARIO	209715200
6	SYSTEM	859832320



## Actividad 2

Crea una tabla llamada vehiculos donde se almacene la matricula, la marca, el modelo, los kilómetros y el precio de vehículos de segunda mano. Utilizar el tipo de campo y la longitud que creáis más adecuados para cada uno de los campos. Introduce datos en la tabla.

Crea un usuario llamado “vendedor” con contraseña “Venta1234” y lo asignas al tablespace concesionario. Este usuario puede iniciar sesión.

El usuario vendedor puede añadir, modificar y consultar datos en la tabla vehiculos.

Comprueba que se han asignado correctamente los permisos.

### Solución:

```
Create table tvehiculos(  
matricula varchar (8) not null,  
marca varchar (10) not null,  
modelo varchar(20) not null,  
kilometros int not null,  
precio decimal(8,2) not null,  
Primary key (matricula));
```

```
insert into tvehiculos values ('1234-ABC','Seat','Leon',12000,10000);  
insert into tvehiculos values ('1234-BCD','Seat','Ibiza',90000,6000);  
insert into tvehiculos values ('1234-EFG','Audi','A3',18200,14000);  
insert into tvehiculos values ('1234-AAA','Opel','Corsa',36000,5000);  
insert into tvehiculos values ('1234-BBB','Ford','Fiesta',45000,7000);
```

```
select * from tvehiculos;
```

	MATRICULA	MARCA	MODELO	KILOMETROS	PRECIO
1	1234-ABC	Seat	Leon	12000	10000
2	1234-BCD	Seat	Ibiza	90000	6000
3	1234-EFG	Audi	A3	18200	14000
4	1234-AAA	Opel	Corsa	36000	5000
5	1234-BBB	Ford	Fiesta	45000	7000

```
create user vendedor identified by "venta1234" default tablespace concesionario quota unlimited  
on concesionario;
```

```
grant create session to vendedor;
```

```
grant insert,update,select on tvehiculos to vendedor;
```



```
SQL> connect vendedor
Introduzca la contraseña:
Conectado.
SQL> select * from joan.tvehiculos;
```

MATRICUL	MARCA	MODELO	KILOMETROS	PRECIO
1234-ABC	Seat	Leon	12000	10000
1234-BCD	Seat	Ibiza	90000	6000
1234-EFG	Audi	A3	18200	14000
1234-AAA	Opel	Corsa	36000	5000
1234-BBB	Ford	Fiesta	45000	7000

```
Select substr(privilege,1,15), substr(table_name,1,15) from dba_tab_privs where
grantee='VENDEDOR';
```

	SUBSTR(PRIVILEGE,1,15)	SUBSTR(TABLE_NAME,1,15)
1	INSERT	TVEHICULOS
2	SELECT	TVEHICULOS
3	UPDATE	TVEHICULOS

### Actividad 3

Crea un rol ventas que pueda con los permisos de conectar, leer, modificar e insertar datos en la tabla de vehículos.

Crea un usuario llamado comercial1 con contraseña com1234 y le asignas el rol ventas. Comprueba que tiene los permisos correspondientes.

Crea un perfil llamado perfilventas que le obligue a cambiar la contraseña cada 30 días. Asigna el perfil al usuario comercial.

#### Solución:

```
Create role rolventas;
Grant create session to rolventas;
Grant select,update,insert on joan.tvehiculos to rolventas;
Create user comercial1 identified by "com1234" default tablespace concesionario quota
unlimited on concesionario;
Grant rolventas to comercial1;
Select grantee, granted_role from dba_role_privs where grantee='COMERCIAL1';
Create profile perfilventas limit
Password_life_time 15;
Alter user comercial1 profile perfilventas;
```