TEMA 9. Lenguaje de Control de Datos SQL.

1. INTRODUCCIÓN.

Ya hemos visto que necesitamos una cuenta de usuario para acceder a los datos de una base de datos. Las claves de acceso se establecen cuando se crea el usuario y pueden ser modificados por el Administrador o por el propietario de dicha clave. La Base de Datos almacena encriptadas las claves en una tabla del diccionario llamada `DBA_USERS`.

La sintaxis es:

```
CREATE USER NombreUsuario
IDENTIFIED BY ClaveAcceso
[DEFAULT TABLESPACE tablespace ]
[TEMPORARY TABLESPACE tablespace]
[QUOTA int {K | M} ON tablespace]
[QUOTA UNLIMITED ON tablespace]
[PROFILE perfil];
```

donde:

- ✔ **CREATE USER**: crea un nombre de usuario que será identificado por el sistema.
- ✔ **IDENTIFIED BY:** permite dar una clave de acceso al usuario creado.
- ✔ **DEFAULT TABLESPACE**: asigna a un usuario el Tablespace por defecto para almacena los objetos que cree. Si no se asigna ninguna, será SYSTEM.
- ✔ **TEMPORARY TABLESPACE**: especifica el nombre del Tablespace para trabajos temporales. Por defecto será SYSTEM.
- ✔ **QUOTA:** asigna un espacio en Megabytes o Kilobytes en el Tablespace asignado. Si no se especifica el usuario no tendrá espacio y no podrá crear objetos.
- ✔ **PROFILE:** asigna un perfil al usuario. Si no se especifica se asigna el perfil por defecto.

Recuerda que para crear usuarios debes tener una cuenta con privilegios de Administrador.

Para ver todos los usuarios creados utilizamos las vistas ALL_USERS y DBA_USERS. Y para ver en mi sesión los usuarios que existen pondría: DESC SYS.ALL_USERS;

Practiquemos un poco con este comando. Creemos una cuenta de usuario limitado, que no tenga derecho ni a guardar datos ni a crear objetos, más tarde le daremos permisos:

```
CREATE USER UsuarioLimitado IDENTIFIED BY passworddemiusuariolimitado ;
```

Podemos modificar usuarios mediante el comando ALTER USER, cuya sintaxis es la siguiente:

```
ALTER USER NombreUsuario
IDENTIFIED BY clave_acceso
[DEFAULT TABLESPACE tablespace ]
[TEMPORARY TABLESPACE tablespace]
[QUOTA int {K | M} ON tablespace]
[QUOTA UNLIMITED ON tablespace]
[PROFILE perfil];
```

Un usuario sin privilegios de Administrador únicamente podrá cambiar su clave de acceso. Para eliminar o borrar un usuario utilizamos el comando DROP USER con la siguiente sintaxis:

```
DROP USER NombreUsuario [CASCADE];
```

La opción CASCADE borra todos los objetos del usuario antes de borrarlo. Sin esta opción no nos dejaría eliminar al usuario si éste tuviera tablas creadas.

2. CONCEDER PERMISOS.

Ningún usuario puede llevar a cabo una operación si antes no se le ha concedido el permiso para ello. En el apartado anterior hemos creado un usuario para iniciar sesión, pero si con él intentáramos crear una tabla veríamos que no tenemos permisos suficientes para ello.

Para poder acceder a los objetos de una base de datos necesitas tener privilegios (permisos). Éstos se pueden agrupar formando roles, lo que simplificará la administración. Los roles pueden activarse, desactivarse o protegerse con una clave. Mediante los roles podemos gestionar los comandos que pueden utilizar los usuarios. Un permiso se puede asignar a un usuario o a un rol.

Un privilegio o permiso se especifica con el comando `GRANT` (conceder).

Si se dan privilegios **sobre los objetos**:

```
GRANT {privilegio_objeto [, privilegio_objeto]...|ALL|[PRIVILEGES]}
ON [usuario.]objeto
TO {usuario1|rol1|PUBLIC} [,{usuario2|rol2|PUBLIC] ...
[WITH GRANT OPTION];
```

donde:

- ✔ **ON** especifica el objeto sobre el que se conceden los privilegios.
- ✔ **TO** señala a los usuarios o roles a los que se conceden privilegios.
- ✔ **ALL** concede todos los privilegios sobre el objeto especificado.
- ✔ **[WITH GRANT OPTION]** permite que el receptor del privilegio se lo asigne a otros.
- ✔ **PUBLIC** hace que un privilegio esté disponible para todos los usuarios.

En el siguiente ejemplo el usuario Juan ha accedido a la base de datos y ejecuta los siguientes comandos:

GRANT INSERT TO Usuarios TO Ana;

permitirá a Ana insertar datos en la tabla Usuarios

GRANT ALL ON Partidas TO Ana;

Juan concede todos los privilegios sobre la tabla Partidas a Ana

Los privilegios de sistema son los que dan derecho a ejecutar comandos SQL o acciones sobre objetos de un tipo especificado. Existen gran cantidad de privilegios distintos.

La sintaxis para dar este tipo de privilegios la tienes aquí:

```
GRANT {Privilegio1 | rol1 } [, privilegio2 | rol2}, ...]
TO {usuario1 | rol1| PUBLIC} [, usuario2 | rol2 | PUBLIC} ... ]
[WITH ADMIN OPTION];
```

Donde

- ✔ **TO** señala a los usuarios o roles a los que se conceden privilegios.
- ✔ **WITH ADMIN OPTION** es una opción que permite al receptor de esos privilegios que pueda conceder esos mismos privilegios a otros usuarios o roles.
- ✔ **PUBLIC** hace que un privilegio esté disponible para todos los usuarios.

Veamos algunos ejemplos:

```
GRANT CONNECT TO Ana;
```

Concede a Ana el rol de CONNECT con todos los privilegios que éste tiene asociados.

```
GRANT DROP USER TO Ana WITH ADMIN OPTION;
```

Concede a Ana el privilegio de borrar usuarios y que ésta puede conceder el mismo privilegio de borrar usuarios a otros.

3. QUITAR PERMISOS.

Hasta ahora hemos aprendido a conceder permisos o privilegios. Será importante aprender a retirarlos. Con el comando REVOKE se retiran los privilegios:

- ✔ **Sobre objetos:**

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

- ✔ **Del sistema o roles a usuarios:**

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

Juan va a quitar el permiso de seleccionar y de actualizar sobre la tabla Usuarios a Ana:

```
REVOKE SELECT, UPDATE ON Usuarios FROM Ana;
```

y va a quitarle el permiso de eliminar usuarios:

```
REVOKE DROP USER FROM Ana;
```