

Un ejemplo de CREATE VIEW es el siguiente:

```
CREATE VIEW nba.jugadoresMiami AS
  SELECT Nombre, Posicion FROM nba.jugadores WHERE Nombre_equipo='HEAT';
SELECT * from nba.jugadoresMiami;
```

Además, se pueden crear vistas para que los usuarios no expertos puedan acceder de forma fácil a la información, proporcionándoles, a través de una vista, información obtenida a través de una sentencia SQL compleja:

```
CREATE VIEW VistaPedidos (CodigoPedido,Cliente,Total) AS
  SELECT CodigoPedido, NombreCliente, SUM(Cantidad*PrecioUnidad)
  FROM Clientes NATURAL JOIN Pedidos NATURAL JOIN DetallePedidos
  GROUP BY CodigoPedido;
```

Para eliminar una vista se hace uso del comando DROP VIEW:

```
DROP VIEW [esquema.]nombre_vista;
```

Hay pequeñas variaciones en la sintaxis de los comandos CREATE VIEW y DROP VIEW dependiendo del SGBD que se utilice. Además, se dispone también de un comando ALTER VIEW para hacer modificaciones a la definición de la vista. Para más información sobre estas variaciones, consultar los manuales de cada gestor.

5.13. Los usuarios

Para crear cuentas de usuario que permitan a los usuarios acceder a ciertos objetos con un nivel determinado de privilegios hay que hacer uso del comando CREATE USER.

```
CREATE USER nombre_usuario IDENTIFIED BY 'password' [opciones];
```

Esta sencilla sentencia crea una cuenta usuario que permite la autenticación de un usuario en el SGBD a través de la password identificada mediante la opción IDENTIFIED BY. Esta sintaxis de create user es válida tanto para MySQL como

para Oracle, aunque a Oracle se le pueden incluir multitud de opciones extras para dar características adicionales, como la asignación de cuota para añadir información a un tablespace o bloquear la cuenta temporalmente. En DB2 no se utiliza el comando CREATE USER puesto que la mayor parte de la gestión de usuarios se hace de forma integrada con el Sistema Operativo. A continuación se muestran ejemplos de creación de usuarios:

```
#creación de usuario en MySQL
CREATE USER paco IDENTIFIED BY 'o99238kjkA';

--creación de usuario en Oracle
CREATE USER paco
  IDENTIFIED BY 'o99238kjkA'
  DEFAULT TABLESPACE 'Nominas'
  QUOTA UNLIMITED ON 'Nominas'
  ACCOUNT LOCK;
```

◊ **Actividad 5.3:** Crea los usuarios Pedro y Javier en los SGBD mysql y Oracle con passwords apropiadas. En Oracle, asigna cuota de 1MB en algún tablespace.

Para eliminar usuarios, se puede utilizar la sentencia DROP USER:

```
DROP USER nombre_usuario [CASCADE];
```

En Oracle, se puede incluir el token CASCADE para indicar que junto con el usuario se borren todos los objetos de su esquema. En MySQL, se pueden borrar a la vez varios usuarios, separando los nombres de los usuarios mediante comas.

Para modificar usuarios, Oracle utiliza el comando ALTER USER, que no está disponible en MySQL. Este comando ALTER USER usa las mismas opciones que CREATE USER.

```
--Modificación de usuario en Oracle
ALTER USER Paco
  IDENTIFIED BY 'nueva_pass' DEFAULT Tablespace 'Facturas';
```

En MySQL hay que modificar los usuarios actualizando sus datos en la propia tabla mysql.user del sistema. Por ejemplo, para cambiar el host desde el que el

usuario Javier se puede conectar, habría que ejecutar la siguiente sentencia. Cuando en MySQL se modifica algún permiso modificando el contenido de las tablas del sistema, hay que ejecutar además el comando FLUSH PRIVILEGES para forzar al gestor a volver leer las tablas de permisos y que los cambios en los permisos sean efectivos desde ese momento.

```
#Modificación de usuario en mysql.  
UPDATE mysql.user  
SET host='192.168.3.1' where user='JAVIER';  
FLUSH PRIVILEGES;
```

En MySQL también se puede renombrar un usuario conservando todos sus privilegios utilizando el comando RENAME USER y cambiar la password mediante el comando SET PASSWORD:

```
#Modificación de usuarios en MySQL. Cambiar nombre y password al usuario Paco  
RENAME USER Paco@localhost to PacoSanchez@localhost;  
SET PASSWORD for Paco@localhost = PASSWORD('nueva_pass');
```

◊ **Actividad 5.4:** Ejecuta el siguiente comando para cambiar la password de un usuario en mysql:

```
UPDATE mysql.user  
SET Password = PASSWORD("nueva_pwd")  
WHERE user='paco' AND host='localhost';
```

◊ **Actividad 5.5:** Cambia la password de los usuarios Pedro y Javier creados en Oracle y bloquea sus cuentas mediante la opción ACCOUNT UNLOCK.

5.14. Los privilegios

Un usuario puede obtener privilegios para manipular objetos de una base de datos con el comando GRANT. Asimismo, se le pueden denegar permisos con el

comando REVOKE. Estos comandos varían en su sintaxis dependiendo del SGBD que se está usando puesto que el sistema de seguridad de cada uno es distinto. Además, influye también la forma en que el subsistema de permisos del SGBD se integra con el sistema operativo.

5.14.1. El sistema de privilegios de MySQL

La sintaxis del comando GRANT para MySQL es la siguiente:

```
GRANT tipo_privilegio [(columnas)] [, tipo_privilegio [(columnas)]] ...
ON {nombre_tabla | * | *.* | base_datos.* | base_datos.nombre_tabla}
TO usuario [IDENTIFIED BY [PASSWORD] 'password']
[, usuario [IDENTIFIED BY [PASSWORD] 'password']] ...
[WITH opcion [opcion] ...

opcion =
    GRANT OPTION
    | MAX_QUERIES_PER_HOUR count
    | MAX_UPDATES_PER_HOUR count
    | MAX_CONNECTIONS_PER_HOUR count
    | MAX_USER_CONNECTIONS count
```

En MySQL se puede otorgar a un usuario permisos para hacer cualquier operación a nivel de host, de base de datos, de tabla o de columna. Así, es posible asignar, por ejemplo, permisos de SELECT sobre las columnas NombreCliente, Dirección y Teléfono de la tabla clientes:

```
GRANT SELECT (NombreCliente,Telefono,Ciudad)
ON Clientes TO paco@localhost;
```

Con esta sentencia el usuario *paco@localhost* solo podrá seleccionar las columnas *NombreCliente*, *Telefono* y *Ciudad* de la tabla clientes, siéndole denegada una consulta del tipo *select * from clientes*.

tipo_privilegio es la clase de permiso que puede ser otorgado, típicamente pueden ser *select*, *insert*, *update*, ... Pueden ser de los más variados. En el Cuadro 5.3 de la página 210 hay algunos ejemplos de los permisos disponibles en MySQL.

Estos tipos de privilegio se pueden aplicar a las siguientes expresiones:

Expresión	Se aplica el permiso a
nombre_tabla	La tabla nombre_tabla
*	Todas las tablas de la base de datos que se está usando
.	Todas las tablas de todas las bases de datos
base_datos.*	Todas las tablas de la base de datos db_name
base_datos.nombre_tabla	Solo la tabla nombre_tabla de la base de datos db_name

Cuadro 5.1: Tipos de objetos a los que se puede otorgar permisos.

Finalmente con 'TO usuario' se indica el usuario al que se quiere otorgar el permiso. Si el usuario user no existe, se crea, opcionalmente con la password indicada mediante la cláusula identified by.

Adicionalmente, se puede indicar ciertas opciones precedidas de la cláusula WITH:

Expresión	Función
GRANT OPTION	Permite conceder a otros usuarios los permisos que tiene el usuario, por tanto, el administrador debe ser muy cauto a la hora de conceder esta opción a los usuarios de la base de datos.
MAX_QUERIES_PER_HOUR count	Permite restringir el número de consultas por hora que puede realizar un usuario.
MAX_UPDATES_PER_HOUR count	Permite restringir el número de modificaciones por hora que puede realizar un usuario.
MAX_CONNECTIONS_PER_HOUR count	Permite restringir las conexiones (logins) por hora que realiza un usuario.
MAX_USER_CONNECTIONS count	Permite limitar el número de conexiones simultáneas que puede tener un usuario.

En cualquiera de las opciones MAX_ si a count se le da el valor 0, significa ilimitado.

Cuadro 5.2: Opciones adicionales a los permisos.

Privilegio	Significado
ALL [PRIVILEGES]	Da todos los permisos simples excepto GRANT OPTION
ALTER	Permite el uso de ALTER TABLE
ALTER ROUTINE	Modifica o borra rutinas almacenadas
CREATE	Permite el uso de CREATE TABLE
CREATE ROUTINE	Crea rutinas almacenadas
CREATE TEMPORARY TABLES	Permite el uso de CREATE TEMPORARY TABLE
CREATE USER	Permite el uso de CREATE USER, DROP USER, RENAME USER, y REVOKE.
CREATE VIEW	Permite el uso de CREATE VIEW
DELETE	Permite el uso de DELETE
DROP	Permite el uso de DROP TABLE
EXECUTE	Permite al usuario ejecutar rutinas almacenadas
FILE	Permite el uso de SELECT ... INTO OUTFILE y LOAD DATA INFILE
INDEX	Permite el uso de CREATE INDEX y DROP INDEX
INSERT	Permite el uso de INSERT
LOCK TABLES	Permite el uso de LOCK TABLES en tablas para las que tenga el permiso SELECT
PROCESS	Permite el uso de SHOW FULL PROCESSLIST
RELOAD	Permite el uso de FLUSH
REPLICATION CLIENT	Permite al usuario preguntar dónde están los servidores maestro o esclavo
REPLICATION SLAVE	Necesario para los esclavos de replicación
SELECT	Permite el uso de SELECT
SHOW DATABASES	SHOW DATABASES muestra todas las bases de datos
SHOW VIEW	Permite el uso de SHOW CREATE VIEW
SHUTDOWN	Permite el uso de mysqladmin shutdown
UPDATE	Permite el uso de UPDATE
USAGE	Sinónimo de 'no privileges', permite únicamente la conexión al gestor
GRANT OPTION	Permite dar permisos

Cuadro 5.3: Tipos de privilegios en mysql.

Algunos ejemplos de consultas para asignación de permisos en MySQL son los siguientes:

```
#Otorga permisos de select e insert a todas las tablas de nba
GRANT SELECT, INSERT on nba.* TO paco@localhost;

#Otorga todos los privilegios a la tabla Clientes de jardineria
GRANT ALL PRIVILEGES on jardineria.Clientes TO paco@localhost;

#Otorga permisos de select a todas las tablas de todas las bases de datos
#permitiendo al usuario ceder esos permisos a otros usuarios
GRANT SELECT on *.* to paco@localhost WITH GRANT OPTION;

#Otorga permisos de SELECT, INSERT, UPDATE y DELETE con un límite de 10
#consultas a la hora en la tabla jugadores de la nba
GRANT SELECT,INSERT,UPDATE,DELETE on nba.jugadores to paco@localhost
WITH MAX_QUERIES_PER_HOUR 10 MAX_UPDATES_PER_HOUR 10
```

La sentencia revoke deniega permisos a un usuario sobre un objeto. A continuación se describe la sintaxis:

```
REVOKE tipo_privilegio [(columnas)] [, tipo_privilegio [(columnas)]] ...
ON {nombre_tabla | * | *.* | base_datos.* | base_datos.nombre_tabla}
FROM usuario [, usuario] ...
```

La sintaxis es muy parecida a la de la sentencia grant, a continuación se muestran unos cuantos ejemplos:

```
#quita el permiso de select en la tabla jardineria.Cliente
revoke select on jardineria.Clientes from paco@localhost;

#elimina el permiso ALL PRIVILEGES de todas las tablas
revoke all privileges on *.* from paco@localhost;

#quita los permisos de select e insert de todas las tablas de jardineria
revoke select,insert on jardineria.* from paco@localhost;
```

Otra forma de asignar y eliminar permisos en MySQL es utilizando las tablas del catálogo de metadatos, es decir, las tablas de la base de datos *mysql* creada en toda instalación del SGBD y que almacena toda la información sobre todos los

objetos manejados por el gestor. En esta base de datos existen 5 tablas relacionadas con el sistema de permisos de MySQL, `user`, `db`, `host`, `tables_priv` y `columns_priv`. Estas tablas se pueden manipular manualmente con inserts y deletes para otorgar y denegar permisos a nivel de usuario, base de datos, equipo, tablas y columnas respectivamente.

A continuación se ilustra cómo MySQL procesa una consulta. Al recibir la instrucción SQL, se comprueba si el acceso a los diversos objetos están autorizados en cualquiera de estas tablas. Si ninguno de los niveles permite el acceso, la consulta es denegada por falta de permisos.

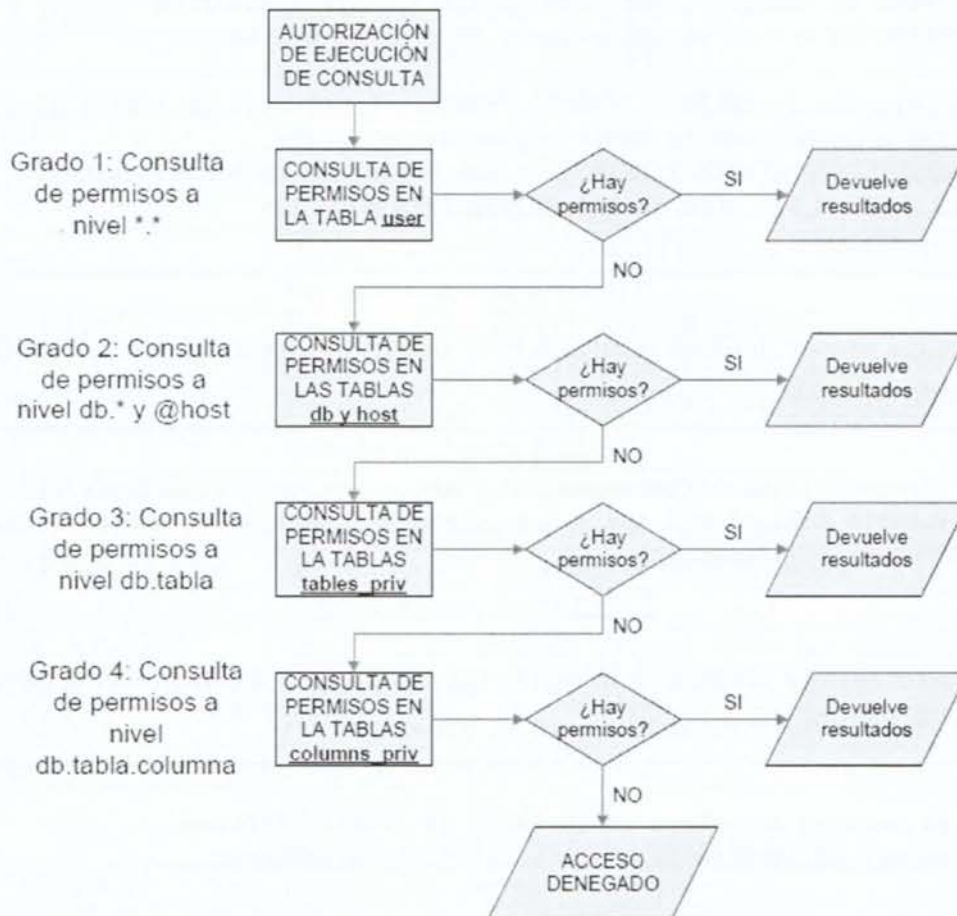


Figura 5.7: Estructura de permisos en mysql.

5.14.2. El sistema de privilegios de Oracle

El sistema de privilegios de Oracle clasifica los permisos en dos tipos, privilegios del sistema y privilegios de objetos. Además define los siguientes conceptos:

- **ROLES** Conjunto de privilegios que se pueden asignar a un determinado usuario. Un usuario puede pertenecer a múltiples roles.

5.16. Prácticas Propuestas

Práctica 5.3: Vincular tablas a través de Access / ODBC

Mediante el driver ODBC para MySQL, enlaza a una BBDD Access las tablas de la NBA y las tablas de la BBDD jardineria. A continuación, realiza las siguientes acciones:

1. Exporta a Excel la tabla jugadores de la NBA.
2. Con Word, crea una carta modelo de felicitación de navidad a los clientes de la base de datos y combina la correspondencia para que, automáticamente, se genere una carta para cada cliente.
3. Inserta dos registros en la tabla Empleados de jardineria mediante un formulario creado en Access, y después, inserta dos jugadores de la NBA siguiendo el mismo procedimiento. Hay que asegurarse de que, efectivamente, están los registros insertados.

Se puede repetir esta misma operación con el driver ODBC para Oracle, pero hay que tener en cuenta que junto al driver, se debe instalar el software cliente de Oracle (sqlplus, tnsnames, etc.) para que funcione. ◇

Práctica 5.4: Actualizaciones y borrados variados

1. Modifica la tabla DetallePedido para insertar un campo numérico llamado IVA. Mediante una transacción, establece el valor de ese campo a 18 para aquellos registros cuyo pedido tenga fecha a partir de Julio de 2010. A continuación actualiza el resto de Pedidos estableciendo al 16 el IVA.
2. Modifica la tabla DetallePedido para incorporar un campo numérico llamado TotalLinea, y actualiza todos sus registros para calcular su valor con la fórmula $\text{TotalLinea} = \text{PrecioUnidad} * \text{Cantidad} * \text{IVA} / 100$.
3. Borra el cliente que menor límite de crédito tenga. ¿Es posible borrarlo solo con una consulta? ¿Por qué?
4. A través de phpMyAdmin o, mediante Access (vinculado vía ODBC), inserta dos clientes nuevos para un empleado a tu elección. A continuación, inserta un pedido con al menos 3 líneas de detalle. Después, ejecuta una consulta para rebajar en un 5 % el precio de los productos que sean más caros de 200 euros. ◇

Práctica 5.6: MySQL: Crear usuarios y asignar permisos en local

Realiza las siguientes operaciones y almacena los comandos y los resultados en un fichero:

1. Crea un usuario llamado `paco@localhost` con la sintaxis `create user` con permisos de solo conexión y comprueba que se pueda conectar.
2. Crea un usuario llamado `juan@localhost` con la sintaxis `grant` con permisos de solo conexión y comprueba que se pueda conectar.
3. Otorga al usuario `paco@localhost` permisos de `select` en la tabla `jardineria.Clientes` y comprueba que se pueda consultar la tabla.
4. Otorga al usuario `juan@localhost` permisos de `select`, `insert` y `update` en las tablas de la base de datos `jardineria` con opción `GRANT`.
5. Conéctate con el usuario `juan` y otorga permisos a `paco` de selección en la tabla `jardineria.Empleados`.
6. Quítale ahora los permisos a `paco` de selección sobre la tabla `jardineria.Clientes`.
7. Conéctate con `root` y elimina todos los permisos que has concedido a `Paco` y `Juan`.
8. Otorga a `juan` los permisos de `SELECT` sobre las columnas `CodigoOficina` y `Ciudad` de la tabla `Oficinas` de la base de datos `jardineria`.
9. Conéctate con `juan` y ejecuta la query `'SELECT * from jardineria.Oficinas'` ¿Qué sucede?.
10. Borra el usuario `paco@localhost`.

◇

Práctica 5.7: MySQL: Crear usuarios y asignar permisos en remoto

Realiza las siguientes operaciones y almacena los comandos y los resultados en un fichero:

1. Crea un nuevo usuario llamado `usuario@direccion_ip` donde `direccion_ip` es una máquina de un compañero tuyo y `usuario` su nombre.
2. Otórgale permisos de selección en todas las tablas de la base de datos `jardineria`. Ten cuidado, es posible que tu servidor solo permita conexiones desde el ordenador local, para permitir conexiones remotas debes comentar la línea `bind-address` de tu fichero `my.cnf` que impide conexiones desde otros sitios que no sea el especificado (`127.0.0.1`). Asegúrate de reiniciar el servidor.
3. Pide a tu compañero que se conecte desde su máquina y que averigüe qué permisos le has otorgado. El a tí te pedirá lo mismo, es decir, que te conectes a su máquina, indica qué instrucción sql ejecutas para conocer los permisos que tienes.
4. Revócale los permisos concedidos al usuario `usuario@direccion_ip`.
5. Concédetele ahora permisos de creación de tablas en una nueva base de datos que has creado.
6. Solicítale que se conecte y que pruebe a crear una tabla. ¿Puede consular la información?.
7. Borra ahora el usuario `usuario@direccion_ip`.
8. Con la bdd `mysql` consulta qué privilegios tiene el usuario `juan@localhost` a nivel de servidor, a nivel de base de datos, a nivel de tablas y a nivel de columnas. Utiliza el comando `show grants for usuario`.

◇