

Tema 11. DCL. Creación y Gestión de usuarios MySQL

1. Introducción.....	1
2. Crear nuevos usuarios.....	1
3. Modificar, renombrar y borrar usuarios.....	2
4. Asignar permisos.....	2
4.1. Permisos.....	3
4.2. Otras sentencias a tener en cuenta.....	3
4.3. Opción WITH GRANT OPTION.....	3
5. Quitar permisos.....	4

1. Introducción.

El servidor de base de datos MySQL permite que más de un usuario pueda trabajar con los recursos del servidor (registros, tablas, bases de datos, funciones, etc..).

Si nosotros somos los únicos administradores del servidor (algo que muy pocas veces sucede) podemos utilizar el usuario **root**, aunque no es recomendable, en ese caso se debe de crear un usuario con permiso de administrador y trabajar con ese usuario.

Sin embargo, si más personas trabajaran con el servidor, será necesario que crear nuevos usuarios y asignarle a cada uno los permisos pertinentes.

2. Crear nuevos usuarios.

La orden SQL que permite añadir nuevos usuarios es: CREATE USER

Su forma más simple:

```
CREATE USER nombre_usuario IDENTIFIED BY 'password';
```

“nombre_usuario” tiene el formato: 'usuario'@'host'

Por ejemplo:

```
CREATE USER 'mercurio'@'localhost' IDENTIFIED BY '123456';
```

Cuando hemos creado el nuevo usuario, también estaríamos añadiendo una nueva fila a la tabla 'user' de la base de datos 'mysql', con nombre de usuario 'mercurio', pudiendo sólo acceder desde 'localhost' y password '123456'.

Cuidado: Si no ponemos la palabra clave 'IDENTIFIED BY' estaríamos creando un usuario sin contraseña, lo cual no es recomendado.

Por ejemplo:

```
CREATE USER 'anonimo'@'localhost';
```

Al crear este usuario sin contraseña, podríamos conectarnos a MySQL sin enviar el parámetro -p o --password en herramientas clientes. Por ejemplo:

```
$mysql -u anonimo
```

Si no indicamos el host, Mysql asignará el patrón '%' al mismo. Por ejemplo:

```
CREATE USER 'anonimo';
```

Es importante tener en cuenta el formato del usuario de la forma: 'usuario'@'host', ya que si ponemos esto: 'usuario@host' estaríamos creando un usuario de nombre 'usuario@host' que puede conectarse desde cualquier PC. Por defecto se le asigna “%” como host.

Para ver la sentencia 'CREATE USER' utilizada para crear un determinado usuario, debemos ejecutar la orden MySQL: SHOW CREATE USER usuario;

Por ejemplo, para ver la sentencia 'create user' del usuario actual podemos poner:

```
mysql>SHOW CREATE USER CURRENT_USER();
```

Hasta este punto, nosotros ya podemos identificarnos con el servidor utilizando el nuevo usuario, sin embargo, una vez identificado las acciones que podemos hacer son mínimas, debido a que este usuario no posee los permisos necesarios para trabajar con las bases de datos.

Solo tenemos permiso de USAGE.

Desde work-bench debemos de crear una nueva conexión con los usuarios dados de alta, y hacer uso de ellas para probar que todo funciona correctamente.

Si estamos trabajando desde terminal, habría que salir (exit;) y volver a conectarnos (mysql -u usuario -p). Aunque podemos utilizar:

```
mysql>\! mysql -u usuario -p;
```

3. Modificar, renombrar y borrar usuarios.

La orden SQL que permite modificar usuarios es: ALTER USER. Entre otros aspectos se puede **cambiar la contraseña**, conservándose el nombre, host y permisos. Un ejemplo sería así:

```
mysql>ALTER USER 'mercurio'@'%' IDENTIFIED BY 'nueva_password';
```

Para **modificar el nombre del usuario** debemos hacer uso de la orden MySQL: RENAME USER.

Un ejemplo:

```
mysql>RENAME USER 'mercurio'@'%' to 'venus'@'localhost';
```

La orden SQL que permite dar de **baja o borrar** a usuarios es: DROP USER. Un ejemplo:

```
mysql>DROP USER 'mercurio'@'%';
```

4. Asignar permisos.

La orden MySQL que permite asignar permisos a los usuarios es: GRANT.

Cuando se crea un usuario el único permiso que tiene es el de “USAGE” que solo le da derecho a conectarse a MySQL, no puede hacer nada más.

Para poder establecer permisos, debemos ser el usuario root u otro usuario que tenga privilegios de asignar permisos.

Si queremos que el nuevo usuario tenga permisos de administrador (Todos los permisos), debemos de ejecutar la siguiente sentencia:

```
mysql>GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'localhost';
```

o

```
mysql>GRANT ALL ON *.* TO 'nombre_usuario'@'localhost';
```

Los asteriscos indican que los permisos serán asignados a todas las bases de datos y a todas las tablas (primer asterisco bases de datos, segundo asterisco tablas).

Si queremos asignar permisos para ciertas acciones, la sentencia quedaría de la siguiente manera. Reemplazamos ALL PRIVILEGES y colocamos las acciones que queremos asignar.

Por ejemplo:

```
mysql>GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON
      MiBBDD.* TO 'nombre_usuario'@'localhost';
```

En esta ocasión estamos indicando que el usuario: 'nombre_usuario'@'localhost' podrá consultar, crear, actualizar y eliminar registros, así cómo podrá crear o eliminar elementos (tablas, índices, columnas, funciones, stores, etc..). Todos estos permisos serán válidos únicamente en la base de datos “MiBBDD” y se aplicarán a todas las tablas.

Si queremos asignar permisos únicamente a una tabla, reemplazamos el asterisco por el nombre de la tabla. Por ejemplo:

```
mysql>GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
      ON MiBBDD.mitabla TO 'nombre_usuario'@'localhost';
```

4.1. Permisos.

Aquí un listado de algunos permisos que podemos asignar.

- **CREATE** permite crear nuevas tablas o bases de datos.
- **DROP** permite eliminar tablas o bases de datos.
- **DELETE** permite eliminar registros de tablas.
- **INSERT** permite insertar registros en tablas.
- **SELECT** permite leer registros en las tablas.
- **UPDATE** permite actualizar registros en las tablas.
- **GRANT OPTION** permite asignar permisos de usuarios.
- **SHOW DATABASES** Permite listar las bases de datos existentes.

4.2. Otras sentencias a tener en cuenta.

Listado de todos los usuarios de mi SGBD:

```
mysql>SELECT User, host FROM mysql.user;
```

Mostrar el usuario con el que te has conectado:

```
SELECT current_user();
```

Mostrar los permisos o privilegios que tiene asignados el usuario de la sesión:

```
SHOW GRANTS;
```

Mostrar los permisos o privilegios que tiene asignados un usuario concreto:

```
SHOW GRANTS FOR 'usuario';
```

4.3. Opción WITH GRANT OPTION.

La orden GRANT tiene la cláusula WITH GRANT OPTION al final de la misma.

Esta opción lo que conlleva es que el permiso que estamos concediendo a un usuario con la orden GRANT, lo va a poder gestionar. Es decir, que va a poder otorgar el mismo permiso a otros usuarios.

Si otorga el permiso a otros usuarios con la opción WITH GRANT OPTION esos otros usuarios van a poder otorgar el mismo permiso a terceros. Y así sucesivamente. Por ejemplo:

```
mysql>GRANT SELECT,INSERT ON *.* TO 'marce'@'%' WITH GRANT OPTION;
```

5. Quitar permisos.

La orden MySQL que permite asignar permisos a los usuarios es: **REVOKE**.

Los permisos que se pueden quitar son los mismos que se dieron con GRANT. Ejemplo, para quitar los permisos: insert y update al usuario: mercurio@'localhost':

```
mysql>REVOKE UPDATE, DELETE ON *.* FROM 'usuario'@'localhost';
```

Quitar todos los privilegios o permisos.

```
mysql>REVOKE ALL PRIVILEGES ON *.* FROM 'usuario'@'localhost';
```

o

```
nmysql>REVOKE ALL ON *.* FROM 'usuario'@'localhost';
```

6. Acceso remoto.

Para permitir que un usuario tenga acceso remoto se han de cumplir dos condiciones:

- Indicar desde que ordenador se accederá. Esto se hace especificandolo en el host (usuario@host) de debe ser la ip desde la que realiza el acceso al servidor. Si queremos permitir que acceda desde cualquier ordenador: '%'.
- Por otra parte, el servidor ha de estar configurado para permitir accesos remotos. Esto se hace editando el fichero de configuración.

En Ubuntu: /etc/mysql/mysql.conf.d/mysql.cnf

Se ha de modificar la variable: bind-address. Por defecto tiene el valor: 127.0.0.1 se ha de cambiar a la IP del servidor por la que se van a atender las peticiones remotas. Si tenemos más de una ip y queremos que atienda remotamente por todas ellas se pondrá: 0.0.0.0. Por tanto quedará:

```
bind-adress=0.0.0.0
```

Una vez realizado esta cambio se requiere reiniciar el servicio, según la configuración que tengamos con uno de los siguientes comandos:

```
$sudo /etc/init.d/mysql restart
```

```
$sudo systemctl restart mysql
```