Profesora: Olga Cuervo Miguélez

TEMA 6

ADMINISTRACIÓN DE BASES DE DATOS. GESTIÓN DE USUARIOS Y PERMISOS EN MYSQL

Módulo: **Bases de Datos**

Ciclo: **DAM**

Introducción

Una de las principales tareas de un administrador de base de datos es la gestión de usuarios. En este temas se describe **cómo gestionar cuentas y permisos para clientes en un servidor MySQL**. Estudiaremos los comandos disponibles para las operaciones básicas habituales (consulta, inserción o creación, eliminación y actualización) sobre cuentas y permisos.

Podemos crear cuentas MySQL:

- ✓ Usando comandos **GRANT**, método recomendado ya que son más conciso y menos propensos a errores.
- ✓ **Manipulando las tablas de permisos** MySQL directamente.
- ✓ A través de **programas** que ofrecen capacidades para administrar el servidor cómo *Workbench* ó *phpMyAdmin.*

Todos los accesos a una base de datos requieren **conexión** a través de un usuario. A los usuarios se les asigna una serie de **privilegios** que les dan permiso para usar ciertos objetos en la base de datos. De esta forma, todo usuario tendrá derecho a utilizar determinados objetos de la base de datos y tendrá restringido el uso de otros.

Para mayor sencillez, la mayoría de los Sistemas de Gestión de Bases de Datos permiten agrupar los permisos que se pueden aplicar a los usuarios en estructuras denominadas **perfiles** y **roles**, que en última instancia son un conjunto de permisos. De esta forma, cuando un usuario quiere conectarse a una base de datos, primero debe autenticarse, es decir, demostrar que es quien dice ser (normalmente mediante una contraseña). Dicha autenticación tendrá asociados privilegios (derechos) y restricciones.

Por tanto, será responsabilidad del administrador crear usuarios y asignarles los diferentes roles y privilegios que les permitan desarrollar su actividad sin comprometer la seguridad de la base de datos.

En general, no es una buena práctica permitir que todos los usuarios con acceso al servidor tengan todos los privilegios. Para preservar la integridad de los datos y estructuras, será conveniente que solo algunos usuarios puedan realizar determinadas tareas, y que otras, que requieren un mayor conocimiento de las estructuras y tablas de las bases de datos, solo puedan ser realizadas por un número limitado y controlado de usuarios.

Gestión de usuarios en MySQL

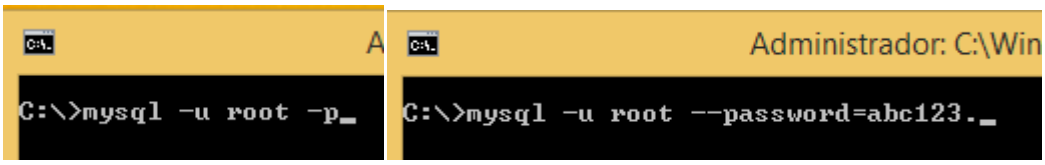
Niveles de privilegios

En MySQL hay cinco niveles diferentes de privilegios:

- ✓ **Global**: se aplican a todas las bases de datos en un servidor. Es el nivel más alto de privilegios, en el sentido de que su alcance es el más general.
- ✓ **Base de datos**: se refieren a bases de datos individuales y, por extensión, a todos los objetos contenidos en cada base de datos.
- ✓ **De tabla**: se aplican a tablas individuales y, por lo tanto, a todas las columnas de esa tabla.
- ✓ **Columna**: aplicar a una columna en una tabla específica.
- ✓ **Rutina**: Se aplican a procedimientos almacenados.

Cómo cualquier operación de administración éste tiene que realizarse por un usuario que tenga permisos administrativos en concreto el permiso **INSERT** sobre la base de datos *mysql* y el permiso **RELOAD**. Para nuestros ejemplos este usuario será el *root*.

En primer lugar, usamos el programa MySQL para conectar al servidor como el usuario *root*, sí hemos asignado una contraseña a la cuenta *root*, necesitaremos la opción *--password* o *–p*



Creación de usuarios

Tras la conexión al servidor como *root*, podemos crear nuevas cuentas. 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**.

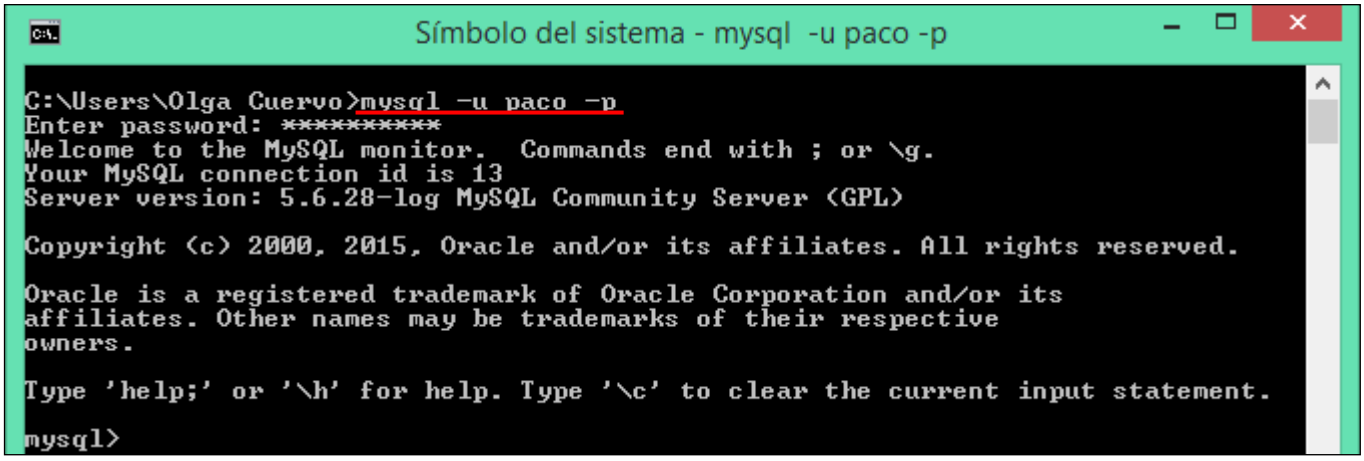
Tutorial: MySQL :: MySQL 8.0 Reference Manual :: 13.7.1.3 CREATE USER Statement

```
CREATE USER usuario [IDENTIFIED BY [PASSWORD] 'contrasinal']
                    [, usuario [IDENTIFIED BY [PASSWORD] 'contrasinal']] ...
```

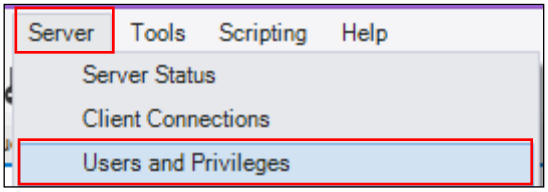
Desde Consola

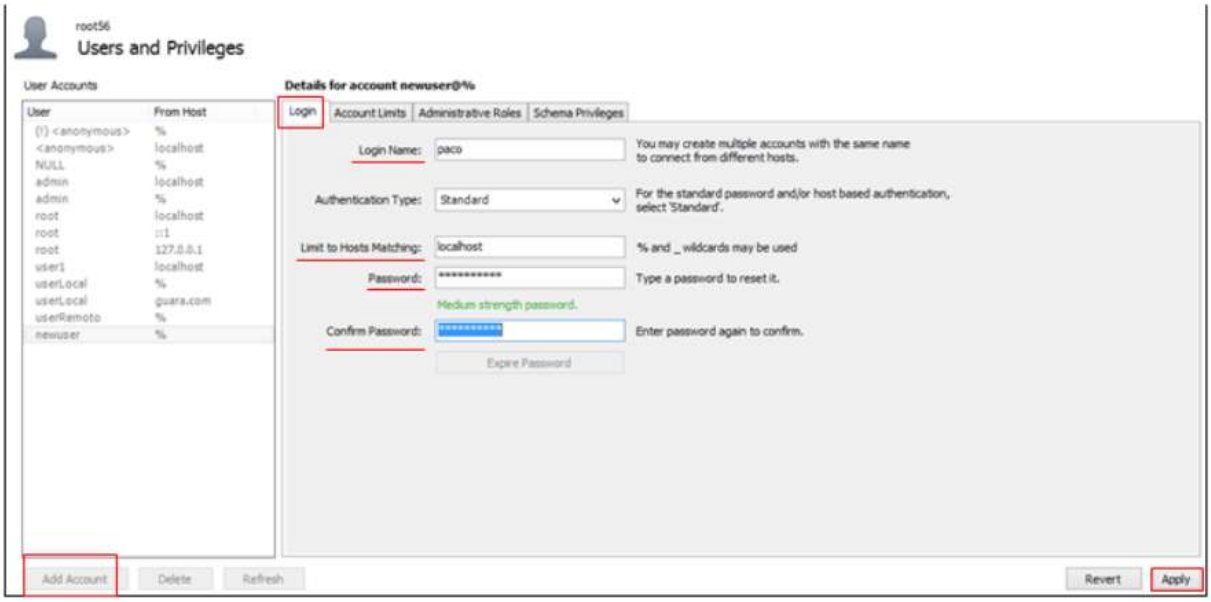
```
mysql> CREATE USER paco@localhost IDENTIFIED BY 'O99238KJkA';
Query OK, 0 rows affected (0.17 sec)
```

Conectamos…

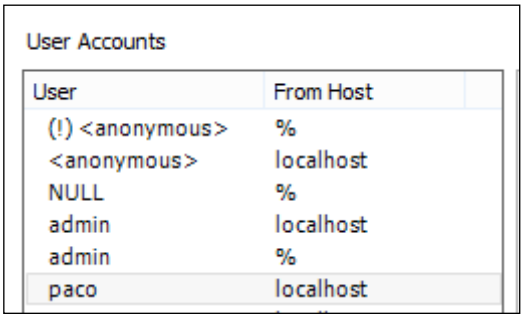


Desde Workbench

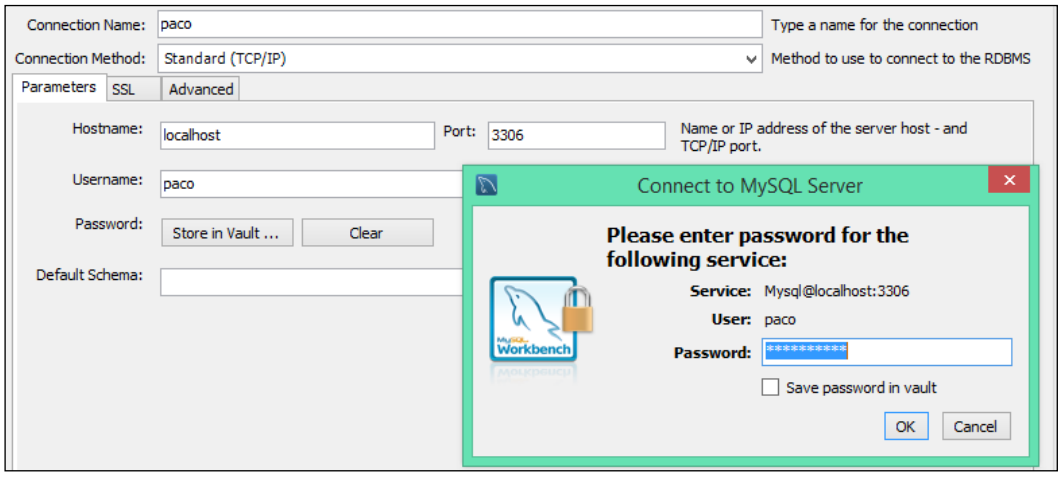




Comprobamos...



Conectamos.....



Usando GRANT puede crear un usuario y al mismo tiempo otorgarle también privilegios, aunque en las últimas versiones de MySQL la recomendación es crear primero el nuevo usuario con CREAR USUARIO y luego usar GRANT para darle privilegios.

La sintaxis simplificada para usar **GRANT** es:

```
GRANT tipo_privilegio [(lista_columnas)] [,tipo_privilegio [(lista_columnas)]] ... ON objeto TO
nombre_usuario[ @equipo ] [ IDENTIFIED BY 'contrasinal' ] [WITH GRANT OPTION]
```

✓ **tipo_privilegio**, representa los privilegios que se pueden otorgar a los usuarios, es decir, qué se le permitirá hacer con los objetos en el servidor. La orden **show privileges** permite ver todos los tipos de privilegios posibles. Algunos de los más usados son:

– ALL [PRIVILEGES]	Todos os privilexios, excepto GRANT OPTION
– ALTER	Modificar obxectos coa orden ALTER (táboas)
– CREATE	Crear obxectos coa orden CREATE (bases de datos ou táboas)
– CREATE VIEW	Crear vistas
– DROP	Borrar táboas con DROP TABLE
– EXECUTE	Executar procedementos almacenados
– SELECT	Facer consultas con SELECT
– UPDATE	Modificar datos das táboas
– USAGE	Sinónimo de 'sen privilexios'
– GRANT OPTION	Conceder privilexios a outros usuarios

✓ En el caso de utilizar la opción **WITH GRANT OPTION**, se le está dando al usuario una posibilidad de transferir a otros usuarios los privilegios otorgados a él.

✓ **objeto**, representa sobre qué cosas se otorgan o retiran privilegios. Los más usados son:

– *.*	Todas as táboas de todas a bases de datos
– *	Todas as táboas da base de datos activa
– nome _bd.*	Todas as táboas da base de datos nome_bd
– nome_db.nome_táboa	A táboa especificada da base de datos nome_bd

También se pueden conceder o retirar privilegiossobre los procedimientos almacenados.

✓ **nombre_usuario**, representa al usuario al que se conceden permisos. El nombre del usuario es una cadena de caracteres que solo deben llevar letras, números y el guión bajo (_).

✓ **equipo**, puede ser un nombre de ordenador, una dirección IP o el símbolo %, que representa cualquier computadora (excepto la máquina local). También se puede utilizar el símbolo % utilizar como carácter comodín en combinación con el nombre del equipo o la dirección IP. Ejemplos de usuarios:

- **'administrador'@'localhost'** Usuario administrador cando se conecte desde o equipo local
- **'administrador'@'%'** Usuario administrador cando se conecte desde calquera equipo da rede
- **'julio'@'ordenador124'** Usuario julio cando se conecta desde o equipo co nome ordenador124
- **'andres'@ '192.68.123.50'** Usuario andres cando se conecta desde o equipo coa IP 192.68.123.50
- **'luis'@'192.68.%.%'** Usuario luis cando se conecta desde un equipo cunha IP que empeza por 192.68

✓ **contraseña**, la cláusula IDENTIFIED BY le permite asignar una contraseña al usuario en el momento en que se conceden los permisos de acceso. Se puede cambiar la contraseña de un usuario usando una de las siguientes sentencias SQL:

```
GRANT USAGE ON *.* TO nombre_usuario[ @equipo ] IDENTIFIED BY 'contraseña';

SET PASSWORD [FOR usuario] = PASSWORD('contraseña');
```

La función **PASSWORD** encripta la contraseña, utilizando el sistema de encriptación de MySQL, que convierte cualquier cadena de texto en una cadena de 41 caracteres en hexadecimal. Existen otros sistemas de encriptación que se pueden utilizar haciendo uso de de otras funciones de cifrado, como SHA1 o MD5.

Si bien siempre se deben conceder los privilegios mínimos necesarios, existen algunos privilegios que son especialmente peligrosos. Por ejemplo, nunca se debe conceder acceso de carácter global. Los siguientes privilegios pueden suponer una amenaza para la seguridad de la base de datos:

▪ Calquera privilexio sobre a base de datos mysql	▪ Nesta base de datos se almacena información de todo o sistema de seguridade do servidor
▪ ALTER	▪ Un usuario podería modificar as táboas de privilexios, e inutilizalas
▪ DROP	▪ Un usuario podería borrar as táboas de privilexios, perdéndose a información das contas, o que impediría o acceso dos usuarios.
▪ FILE	▪ Os usuarios poderían crear un ficheiro con información das contas de usuario, que todo o mundo poda ler.
▪ GRANT	▪ Permite que un usuario poda ceder os seus privilexios a outros usuarios, que poden non ser tan fiables como el
▪ PROCESS	▪ As consultas realizadas poden ser vistas en modo texto, o que inclúe calquera que cambie ou defina contrasinais.
▪ SHUTDOWN	▪ Os usuarios con este privilexio poden parar o servidor, e deixar ao resto dos usuarios sen servizo

Por ejemplo, para crear un usuario sin privilegios:

```
GRANT USAGE ON *.* TO anonimo IDENTIFIED BY 'clave';
```

Hay que tener en cuenta que la contraseña debe ingresarse entre comillas de forma obligatoria. El usuario 'anonimo' podrá abrir una sesión de MySQL usando el comando:

```
mysql -h localhost -ou anonimo -p
```

Pero no podrá hacer mucho más, ya que no tiene privilegios. No tendrá, por ejemplo, oportunidad de hacer selecciones de datos, crear bases de datos o tablas, insertar datos, etc.
Más ejemplos:

- ✓ Conceder permiso para executar os comandos *insert* y *delete* en la tabla *titles* a *Mary* :

```
grant insert, delete

on titles

to mary;
```

- ✓ Conceder permiso para ejecutar el comando *update* en las columnas *price* y *advance* de la tabla *titles* a *public*.

```
grant update (price, advance)

on titles

to public;
```

- ✓ Conceder permiso a *Mary* y *John* para utilizar los comandos *create database* y *create table*.

```
grant create database, create table

to mary, john;
```

- ✓ Conceder todos los permisos de acceso a la tabla *titles* a todos los usuarios.

```
grant all on titles

to public;
```

- ✓ Conceder permiso a *Mary* para utilizar el comando *update* en la tabla *authors* e para conceder ese permiso a outros.

```
 grant update on authors

to mary

with grant option;
```

Revocar privilegios

Para revocar privilegios se usa la sentencia **REVOKE**.

```
REVOKE priv_type [(column_list)] [, priv_type [(column_list)]] ...

ON
```

```
FROM user [, user] ...
```

La sintaxis es similar a la de **GRANT**, por ejemplo, para revocar el privilegio SELECT de la tabla *gente* de la base de datos *proba* al usuario *anonimo*, se usará la sentencia:

```
REVOKE SELECT ON proba.xente FROM anonimo;
```

Mostrar los privilegios de un usuario

Los privilegios otorgados a un usuario se pueden ver usando la cláusula SHOW GRANTS. El resultado de esta sentencia es una lista de sentencias GRANT que se ejecutarán para otorgar los privilegios que tiene el usuario. Por ejemplo:

```
mysql> SHOW GRANTS FOR anonimo;
-----------------------------------------------------------------------
| Grants for anonimo@% |
-----------------------------------------------------------------------
| GRANT USAGE ON *.* TO 'anonimo'@'%' IDENTIFIED BY PASSWORD '*5...' |
| GRANT SELECT ON `proba`.`xente` TO 'anonimo'@'%' |
-----------------------------------------------------------------------
```

Borrar usuarios

Para eliminar usuarios se usa la sentencia **DROP USER**.
Por ejemplo:

```
mysql> DROP USER anonimo;

Query OK, 0 rows affected (0.00 sec)
```

ÍNDICE