(2.9) APÉNDICE: usuarios y privilegios en MySQL

(2.9.1) cuentas de usuario en MySQL

En MySQL el nombre de un usuario está compuesto por el nombre seguido del signo @ y después el ordenador desde el que dicho usuario se conecta, porque se asume que no es lo mismo el usuario ***pepe@192.168.1.35*** que ***pepe@192.168.1.36***, es decir hay diferentes usuarios de nombre ***pepe*** y tendrán por tanto diferentes privilegios según de qué ***pepe*** hablemos en base a la máquina o la red desde la que se conectan.

A partir de esta idea, cuando un usuario se conecta primero se comprueba si tiene permiso para hacerlo (suponiendo que la contraseña sea correcta). Después cada operación que intenta realizar será controlada para saber si se permite o no.

La tabla **mysql.user**, es decir: tabla **user** de la base de datos **mysql** que contiene la información sobre los usuarios de mysql. En ella se observa una de las particularidades de MySQL, los usuarios usan un nombre seguido del host. De esa forma dos usuarios pueden parecer iguales pero al variar la parte del host, se convierten en dos usuarios diferentes. Por ejemplo, usuario@192.168.1.10 sería diferente de usuario@192.168.1.11

En MYSQL el nombre de usuario debe de cumplir:

- Tener un máximo de 16 caracteres
- Debe comenzar por letra
- No puede repetirse para el mismo host
- Si el usuario lleva espacios en blanco, se coloca entre comillas simples

Para la parte que se refiere al host, es posible usar:

- Direcciones IP, como ***192.168.1.10***
- Nombres de host, como ***localhost*** o cualquier otro nombre reconocido por nuestro servidor DNS
- Direcciones IP con máscara, como ***192.168.1.0/255.255.255.0*** (sólo son válidas de 8,16,24 o 32 bits)
- Direcciones IP con el comodín ***%,*** por ejemplo ***192.168.%.%*** representa cualquier máquina de la red 192.168.0.0
- En los dos últimos puntos anteriores, el host se escribe entre comillas simples.

Cada usuario tiene asociada una contraseña así como una serie de operaciones posibles para realizar.

(2.9.2) creación de usuarios

Desde la versión **5.0.2** de MySQL es posible utilizar el comando estándar **CREATE USER**. La sintaxis es:

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

Se pueden crear usuarios sin indicar contraseñas. El nombre de usuario debe incluir el host (como se comentó en el apartado anterior), de otro modo usará el host '%' que representa a cualquier máquina (es un usuario global).

No es obligatorio el apartado IDENTIFIED BY que permite indicar la contraseña; si no se hace uso de él, la contraseña del usuario queda en blanco (situación nada recomendable). La contraseña se puede indicar en texto plano o a través de la función PASSWORD indicado el resultado de aplicar la función PASSWORD() (de esa forma se oculta el texto plano).

Los usuarios así creados no tienen privilegios asociados,

(2.9.3) borrado de usuarios

Se realiza mediante la instrucción:

```
DROP USER usuario [,...];
```

Si el usuario tiene sesión abierta, no se cierra la sesión. Se aplicará el comando al cierre de la sesión de dicho usuario.

(2.9.4) consulta de los usuarios de MySQL

La tabla mysql.user contiene la lista completa de usuarios. Modificar esta tabla permite modificar los usuarios, por lo que las instrucciones INSERT, DELETE o UPDATE en esta tabla añaden, modifican o eliminan usuarios; aunque no se recomienda ni INSERT ni DELETE (al existir las instrucciones de creación y eliminación de usuario estándares).

Las principales columnas de esa tabla son user, host y password.

(2.9.5) modificar usuarios de MySQL

El comando UPDATE sobre la tabla de usuarios, mysql.user, es la forma habitual de hacerlo, pero necesitamos invocar al comando FLUSH PRIVILEGES para que los cambios se realicen al instante. Ejemplos:

```
UPDATE mysql.user SET host='192.168.1.%' WHERE user='opersys';
UPDATE mysql.user SET password=PASSWORD('123456')
WHERE user='clara';
FLUSH PRIVILEGES;
```

(2.9.6) cambiar de nombre a un usuario

Se usa el comando no estándar, RENAME USER, de esta forma:

```
RENAME USER nombreAntiguo TO nombreNuevo
   [,nombreAntiguo2 TO nombreNuevo2 [,...]
```

(2.9.7) concesión de privilegios en MySQL

En MySQL es el comando estándar GRANT el que permite la concesión de privilegios. La sintaxis es extensa:

```
GRANT tipoDePrivilegio[(listaColumnas1)][, tipoDePrivilegio[(listaColumnas2)][,...]]
  ON [tipoDeObjeto]{tabla | * | *.* | baseDeDatos.* | baseDeDatos.tabla}
  TO usuario1 [IDENTIFIED BY [PASSWORD] 'contraseña'][, usuario2...]
  [WITH opción [,opción2[,...]]]
```

El *tipoDeObjeto* puede ser:

- TABLE
- FUNCTION
- PROCEDURE

Si no se indica tipo de objeto, se entiende que nos referimos a una tabla (que es lo habitual.

Las *opciones* del apartado WITH son:

- GRANT OPTION. Que permite que el usuario al que se le conceden los privilegios pueda, a su vez, concederles a otros.
- MAX_QUERIES_PER_HOUR *n*. Permite indicar el máximo número de consultas (*n*) a la hora que se le permiten al usuarios.
- MAX_UPDATES_PER_HOUR *n* Máximo número de operaciones de modificación de datos permitidas en una hora.
- MAX_CONNECTIONS_PER_HOUR *n*. Máximo número de conexiones que se le permiten hacer al usuario en una hora.
- MAX_USER_CONNECTIONS *n*. Conexiones concurrentes que como máximo el usuario puede mantener abiertas.

En todas las opciones anteriores si se le da a *n* un valor de *0*, entonces se traduce como **ilimitado**. Es decir MAX_USER_CONNECTIONS o permite tener ilimitadas conexiones simultáneas.

El objeto al que se le aplican los privilegios puede ser:

- **Una tabla de la base de datos actual**. De ella se indica el nombre simplemente.
- **Una tabla de una base de datos concreta**. Se indica la base de datos y la tabla separadas por un punto.
- **Todas las tablas de la base de datos actual**. Se indica un asterisco.
- **Todas las tablas de una base de datos**. Se indica el nombre de la base de datos seguida de un punto y un asterisco.
- **Todas las tablas de todas las bases de datos**. Dos asteriscos separados por un punto.

Además existen cinco niveles de aplicación de los permisos:

- **Nivel global**. Se aplican a todas las bases de datos. Por lo que se deben aplicar obligatoriamente al objeto ***.*** Ejemplos de privilegios que se aplican obligatoriamente a este nivel son **CREATE USER** o **SHUTDOWN**
- **Nivel de bases de datos**. Se aplican a todos los objetos de una base de datos. Se deben de aplicar usando el asterisco en las tablas pero un nombre de base de datos.
- **Nivel de tabla**. Se aplican sólo a una tabla.
- **Nivel de columna**. Se aplican a columnas de una tabla
- **Nivel de rutina**. Son privilegios que se aplican a rutinas.

La cláusula **IDENTIFIED BY** permite incluso crear el usuario al que se da permisos; de hecho antes de la versión 5 de MySQL era la forma de crear usuarios.

Los posibles permisos (apartado ***tiposDePrivilegio*** de la sintaxis de GRANT) que se pueden otorgar son:

Permiso	Significado
ALL [PRIVILEGES]	Otorga todos los privilegios
ALTER	Permite el uso de **ALTER TABLE**
ALTER ROUTINE	Modifica o borra rutinas almacenadas
CREATE	Permite crear tablas
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 ALL** PRIVILEGES
CREATE VIEW	Permite crear vistas
DELETE	Permite eliminar filas de tablas
DROP	DROP Permite el uso de **DROP TABLE**
EXECUTE	Permite ejecutar rutinas
FILE	Permite importar y exportar datos mediante **SELECT ... INTO OUTFILE** y **LOAD DATA INFILE**
INDEX	Permite crear y borrar índices
INSERT	Permite añadir filas
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

Permiso	Significado
REPLICATION SLAVE	Necesario para los esclavos de replicación (para leer eventos del log binario desde el maestro)
SELECT	Permite consultar datos
SHOW DATABASES	Permite consular la lista completa de bases de datos
SHOW VIEW	Permite el uso de SHOW CREATE VIEW
SHUTDOWN	Permite el uso de mysqladmin shutdown, que cierra la instancia de MySQL
SUPER	Permite el uso de los comandos CHANGE MASTER, KILL, PURGE MASTER LOGS, y SET GLOBAL; además se permite al usuario que el comando mysqladmin debug le permita conectar (una vez) incluso si se ha superado el número máximo de conexiones.
UPDATE	Permite la modificación de datos
USAGE	Sin privilegios, el estado que tiene un usuario que se acaba de crear con CREATE USER.
GRANT OPTION	Permite dar permisos

Ejemplos:

```
GRANT INSERT,DELETE,UPDATE,SELECT ON almacen.pedidos TO ana;
#Permite a la usuaria ana@% permisos de modificación adición ,
#borrado y consulta sobre la tabla pedidos de la base de datos almacen
GRANT INSERT,DELETE,UPDATE,SELECT ON almacen.pedidos TO mario
WITH GRANT OPTION;
#Igual que la anterior para el usuario mario@% que además podrá el
#mismo conceder esos permisos
GRANT SELECT ON almacen.* TO felipe@192.168.1.32;
#al usuario indicado se le permite consultar todas las tablas del almacen
GRANT ALL ON almacen.* TO clara;
#clara puede hacer cualquier operación sobre la tabla de almacenes
GRANT CREATE ON almacen.* TO julian IDENTIFIED BY 'Caswq1209';
#Crea o modifica (si existe) el usuario Julian con la contraseña
indicada y permiso de  creación de tablas en la base de datos almacén
```

(2.9.8) revocación de permisos

El comando estándar REVOKE se encarga de quitar los permisos concedidos a un usuario concreto. La sintaxis del comando es:

```
REVOKE tipoDePrivilegio [(listaColumnas1)]
    [,tipoDePrivilegio[(listaDeColumnas2)]] [,...]
```

```
ON [tipoDeObjeto]{tabla | * | *.* | baseDeDatos.* | baseDeDatos.tabla}
FROM usuario1[, usuario2[,...]]
```

Ejemplo:

```
REVOKE SHUTDOWN ON *.* FROM alberto@localhost;
REVOKE INSERT,DELETE,UPDATE ON almacen.pedidos FROM clara;
REVOKE ALL PRIVILEGES, GRANT OPTION FROM ana, elena;
```

La última instrucción elimina todos los privilegios.

(2.9.9) mostrar información sobre usuarios y privilegios

La información sobre usuarios y privilegios se encuentra, fundamentalmente en las tablas **user** (lista de usuarios), **host** (lista de hosts) y **db** (bases de datos del sistema). La instrucción **DESCRIBE** seguida del nombre de la tabla (por ejemplo *mysql.db*) permite observar las columnas de la tabla y así saber qué consultar

en ellas.

Otras tablas interesantes son:

- **tables_priv**. Privilegios concedidos a las tablas, de cada tabla aparecen los usuarios que pueden operar con ella y los privilegios concretos que se les ha concedido.
- **columns_priv.** Privilegios concedidos a las columnas.

Todas las tablas anteriores se pueden manipular para conceder o quitar permisos, aunque no es muy lógico que esas operaciones se hagan así.