

UNIDAD 7

USUARIOS Y EXTENSIONES (MYSQL)

BASES DE DATOS 22/23
CFGS DAW

BOLETÍN EJERCICIOS 7.1. GESTIÓN DE USUARIOS (MYSQL)

Revisado y ampliado por:

Abelardo Martínez y Pau Miñana

Autor:

Sergio Badal

Licencia Creative Commons



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

ÍNDICE DE CONTENIDO

1. EJERCICIO 1.....	3
2. EJERCICIO 2.....	3
3. EJERCICIO 3.....	3
4. EJERCICIO 4.....	4
5. EJERCICIO 5.....	4

UD7.1. BOLETÍN GESTIÓN DE USUARIOS

Para los cuatro primeros ejercicios, utiliza cualquier base de datos que tengas ya creada, accede a ella desde la consola con un usuario con permisos de superadministrador y ejecuta las sentencias necesarias para conseguir estos hitos. Para el último, ejecuta el script que te proporcionamos.

1. EJERCICIO 1

Crea un nuevo usuario superadministrador, llamado **gandalf** y con la clave 112233, dale con todos los privilegios posibles cuando acceda desde la misma máquina dónde esté instalada la base de datos. ¡Cuidado, este usuario debe poder conceder privilegios!

Sal de la consola (quit), vuelve a entrar con ese nuevo usuario, comprueba que está en la tabla dónde se almacenan los pares **usuario+host**, y cambia su clave por 445566.

Si tienes problemas con SYSTEM_USER ejecuta esta orden desde tu usuario root:

```
GRANT SYSTEM_USER ON *.* TO root;
CREATE USER 'Gandalf'@'%' IDENTIFIED BY '112233';
GRANT ALL PRIVILEGES ON *.* TO 'Gandalf'@'%'; FLUSH PRIVILEGES;
mysql -u Gandalf -p;
ALTER USER 'Gandalf'@'%' IDENTIFIED BY '445566';
SELECT user,host FROM mysql.user
WHERE user = 'Gandalf';
```

2. EJERCICIO 2

Accediendo desde la misma cuenta **gandalf**, crea tres usuarios nuevos llamados **frodo**, **sam** y **gollum**, con las claves que quieras de manera que podamos gestionar los accesos de los tres según accedan desde localhost, o desde cualquier IP.

Después, dale permisos de lectura (SELECT) a todos ellos sobre todas las bases de datos y todas las tablas si acceden desde fuera y de lectura y actualización (UPDATE) si lo hacen desde la máquina dónde está la base de datos.

Bueno, mejor quítale el permiso de actualización sobre todo a **gollum** :-)

Muestra luego los permisos de cada uno de ellos, para los dos tipos de acceso, para asegurarte de que lo has dejado todo como te pedíamos.

3. EJERCICIO 3

```
CREATE USER 'Frodo'@'localhost' IDENTIFIED BY 'samvalentin';
CREATE USER 'Sam'@'localhost' IDENTIFIED BY 'samvalentin';
CREATE USER 'Gollum'@'localhost' IDENTIFIED BY 'mitesoros';
CREATE USER 'Frodo'@'%' IDENTIFIED BY 'samvalentin1';
CREATE USER 'Sam'@'%' IDENTIFIED BY 'samvalentin1';
CREATE USER 'Gollum'@'%' IDENTIFIED BY 'mitesoros1';
```

```
GRANT SELECT, UPDATE
ON *.*
TO 'Frodo'@'localhost', 'Sam'@'localhost', 'Gollum'@'localhost';
GRANT SELECT
ON *.*
TO 'Frodo'@'%', 'Sam'@'%', 'Gollum'@'%';
REVOKE UPDATE
ON *.*
FROM 'Frodo'@'localhost', 'Sam'@'localhost', 'Gollum'@'localhost';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'Gollum'@'localhost';
```

Crea un nuevo usuario llamado **galadriel**, que pueda acceder todas a las tablas de la base de datos de música para leer, actualizar, insertar y borrar cuando accede desde la misma máquina que la base de datos, y únicamente a la tabla libro de la base de datos biblioteca cuando accede desde cualquier lugar.

Muestra luego los permisos de cada uno de ellos, para los dos tipos de acceso, para asegurarte de que lo has dejado todo como te pedíamos.

```
CREATE USER 'Galadriel'@'localhost';
GRANT SELECT, UPDATE, INSERT, DELETE
ON musica.*
TO 'Galadriel'@'localhost';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'Galadriel'@'localhost';
```

```
CREATE USER 'Galadriel'@'%';
GRANT SELECT, UPDATE, INSERT, DELETE
ON biblioteca.libro
TO 'Galadriel'@'%';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'Galadriel'@'%';
```

4. EJERCICIO 4

Lista todos los pares usuario+host que has creado.

Revoca todos los permisos de **frodo**, **sam**, **gollum** y **galadriel** que has concedido dependiendo de sus accesos, en orden inverso a su concesión.

Elimina los usuarios+host.

```
REVOKE SELECT, UPDATE, INSERT, DELETE ON biblioteca.libro FROM 'Galadriel'@'%';
REVOKE SELECT, UPDATE, INSERT, DELETE ON musica.* FROM 'Galadriel'@'localhost';
REVOKE SELECT ON *.* FROM 'Frodo'@'%', 'Sam'@'%', 'Gollum'@'%';
REVOKE SELECT ON *.* FROM 'Frodo'@'localhost', 'Sam'@'localhost', 'Gollum'@'localhost';

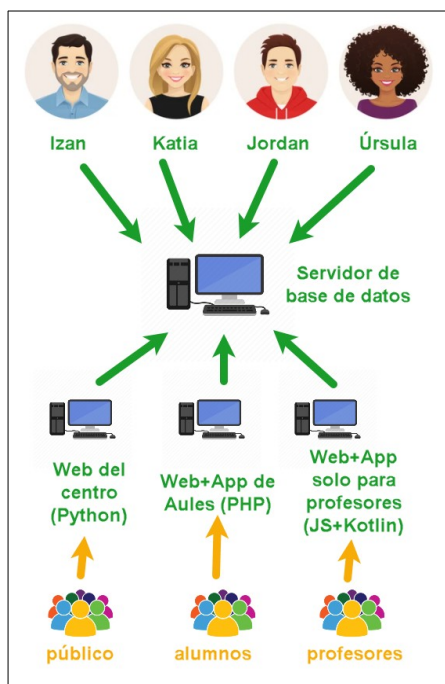
DROP USER 'Galadriel'@'%';
DROP USER 'Galadriel'@'localhost';
DROP USER 'Gollum'@'%';
DROP USER 'Gollum'@'localhost';
DROP USER 'Sam'@'%';
DROP USER 'Sam'@'localhost';
DROP USER 'Frodo'@'%';
DROP USER 'Frodo'@'localhost';
DROP USER 'Gandalf'@'%';
```

5. EJERCICIO 5

VE A LA PÁGINA SIGUIENTE



EJEMPLO AVANZADO GESTIÓN DE USUARIOS MYSQL



Queremos crear los usuarios y permisos necesarios para modelar el acceso a la base de datos de un centro educativo que tiene su propia base de datos, a la que acceden varios técnicos, la web del centro, la web+App de Aules y una web+App creadas para profesores.

El servidor de BD tiene esta información:

- BD alumnado. Tablas: estudiantes, materias
- BD profesorado. Tablas: docentes, documentos
- BD sistema. Tablas: eventos, matriculas, estudios

Cada técnico debe poder acceder desde su casa (con cualquier software que considere o accediendo por consola de manera remota) a cualquier tabla para lectura (SELECT), y a cualquier tabla para SELECT, INSERT, UPDATE y DELETE desde el Dpto. de Informática.

Solo uno de ellos, Úrsula, debe poder conectarse desde el propio servidor (con permiso total) y dar permisos de

todo tipo a todos los usuarios y, solo uno de ellos, Jordan, debe poder conectarse desde cualquier PC del mundo a la tabla sistema.eventos solo para lectura.

Respecto a la web del centro, se podrá conectar usando Python solo a la tabla estudios de la bd sistema y solo en modo lectura.

Respecto a Aules, se podrá conectar usando PHP a todas las tablas de la bd alumnado para SELECT, INSERT, UPDATE y DELETE.

Respecto a la web/App para profesores, se podrá conectar usando JS y/o Kotlin a todas las tablas de la bd profesorado para SELECT, INSERT, UPDATE y DELETE.

El servidor de base de datos está en la IP 254.258.25.4, el Departamento de Informática en la IP 254.258.25.2 los técnicos se conectan desde sus casas con las siguientes IPs: Izan 214.28.5.4, Katia 224.18.5.40, Jordan 254.8.5.22, Úrsula 224.8.53.54. La web del centro está alojada en la IP 254.258.25.1, la app y web de profesores en la 254.258.25.3 y aules en 211.11.12.13.

CONSEJO: Usa la clave '123' en todos los perfiles de usuario para hacer el ejercicio más sencillo.

NOTAS FINALES:

- El equipo directivo cambia de opinión a última hora, y decide quitar el acceso de borrado de Aules a la bd alumnado y de la web/App de profesores a la bd profesorado.
- Izan cambia la IP de su casa a la 214.28.5.6.
- Úrsula cambia su clave para cuando se conecta desde el servidor de la BD: 983422nsd78

>Ejecuta el script que tienes en la siguiente página y prueba tu solución en tu propio servidor<

```

DROP DATABASE IF EXISTS alumnado;
DROP DATABASE IF EXISTS profesorado;
DROP DATABASE IF EXISTS sistema;
CREATE DATABASE alumnado;
CREATE DATABASE profesorado;
CREATE DATABASE sistema;
USE alumnado;
CREATE TABLE estudiantes (id INT PRIMARY KEY, nombre VARCHAR(50));
CREATE TABLE materias (id INT PRIMARY KEY, nombre VARCHAR(50));
CREATE TABLE notas (id INT PRIMARY KEY, nombre VARCHAR(50));
USE profesorado;
CREATE TABLE docentes (id INT PRIMARY KEY, nombre VARCHAR(50));
CREATE TABLE documentos (id INT PRIMARY KEY, nombre VARCHAR(50));
USE sistema;
CREATE TABLE eventos (id INT PRIMARY KEY, nombre VARCHAR(50));
CREATE TABLE matriculas (id INT PRIMARY KEY, nombre VARCHAR(50));
CREATE TABLE estudios (id INT PRIMARY KEY, nombre VARCHAR(50));

```

//Creacion de los usuarios

```

CREATE USER 'Izan'@'214.28.5.4' IDENTIFIED BY '123'; //ip de casa
CREATE USER 'Izan'@'254.258.25.2' IDENTIFIED BY '123'; //ip de dto.informatica

```

```

CREATE USER 'Katia'@'224.18.5.40' IDENTIFIED BY '123'; //ip de casa
CREATE USER 'Katia'@'254.258.25.2' IDENTIFIED BY '123'; //ip de dto.informatica

```

```

CREATE USER 'Jordan'@'254.8.5.22' IDENTIFIED BY '123'; //ip de casa
CREATE USER 'Jordan'@'254.258.25.2' IDENTIFIED BY '123'; //ip de dto.informatica
CREATE USER 'Jordan'@'%' IDENTIFIED BY '123'; //desde cualquier pc del mundo

```

```

CREATE USER 'Ursula'@'224.8.53.54' IDENTIFIED BY '123'; //ip de casa
CREATE USER 'Ursula'@'254.258.25.2' IDENTIFIED BY '123'; //ip de dto.informatica
CREATE USER 'Ursula'@'254.258.25.4' IDENTIFIED BY '123'; //ip de host

```

```

CREATE USER 'Web'@'254.258.25.1' IDENTIFIED BY '123'; //ip de la web

```

```

CREATE USER 'Aules'@'211.11.12.13' IDENTIFIED BY '123'; //ip de aules

```

```

CREATE USER 'App'@'254.258.25.3' IDENTIFIED BY '123'; //ip de la web/app de los profesores

```

//Creacion de permisos

```

GRANT SELECT ON *.* TO 'Izan'@'214.28.5.4','Katia'@'224.18.5.40','Jordan'@'254.8.5.22','Ursula'@'224.8.53.54';
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'Izan'@'254.258.25.2','Katia'@'254.258.25.2','Jordan'@'254.258.25.2','Ursula'@'254.258.25.2';
GRANT ALL PRIVILEGES ON *.* TO 'Ursula'@'254.258.25.4' WITH GRANT OPTION;
GRANT SELECT ON sistema.eventos TO 'Jordan'@'%';
GRANT SELECT ON sistema.estudios TO 'Web'@'254.258.25.1';
GRANT SELECT, INSERT, UPDATE, DELETE ON alumnado.* TO 'Aules'@'211.11.12.13';
GRANT SELECT, INSERT, UPDATE, DELETE ON profesorado.* TO 'App'@'254.258.25.3';

```

//Quitar permisos

```

REVOKE DELETE ON alumnado.* FROM 'Aules'@'211.11.12.13';
REVOKE DELETE ON profesorado.* FROM 'App'@'254.258.25.3';

```

//Cambio de ip de Izan

```

CREATE USER 'Izan'@'214.28.5.6' IDENTIFIED BY '123';
GRANT SELECT ON *.* TO 'Izan'@'214.28.5.6';
REVOKE SELECT ON *.* FROM 'Izan'@'214.28.5.4';

```

//Cambio de contraseña de Ursula

```

ALTER USER 'Ursula'@'254.258.25.4' IDENTIFIED BY '983422nsd78';

```