

**BASES DE DATOS**

**Desarrollo de Aplicaciones Multiplataforma  
Desarrollo de Aplicaciones Web  
GESTIÓN DE BASES DE DATOS**

**Administración de Sistemas Informáticos en Red**

**USUARIOS Y PERMISOS**

**Luis Dorado**

**Vanesa Martínez**

**Pablo Bahillo**

**Alba Tortosa**

Ejercicios

Contenido

[1 Gestión de usuarios 2](#_Toc72146175)

[1.1 SQL 2](#_Toc72146176)

[1.1.2 Ampliación 2](#_Toc72146177)

[1.2 MySQLWorkBench 2](#_Toc72146178)

[1.2.2 Ampliación 2](#_Toc72146179)

[2 GESTIÓN DE PERMISOS 3](#_Toc72146180)

[2.1 GRANT 3](#_Toc72146181)

[2.1.2 Ampliación 3](#_Toc72146182)

[2.2 GRANT / REVOKE / WITH GRANT OPTION 3](#_Toc72146183)

[2.3 ROLES 4](#_Toc72146184)

[3 AMPLIACIÓN: NIVELES DE SEGURIDAD 5](#_Toc72146185)

# Gestión de usuarios

## SQL

#### Crea tres usuarios de nombre u1, u2, u3 utilizando una única instrucción SQL. Todos se van a conectar desde la red 170.10.0.0/16.

* Haz que tengan que cambiar la contraseña cuando se conecten, y que como máximo puedan realizar 50 operaciones de modificación por hora contra la base de datos.
* Sólo podrán conectarse desde un computador simultáneamente.

#### Crea dos usuarios utilizando una única orden SQL, de tal forma que las dos cuentas estén inicialmente bloqueadas

#### Desbloquea las cuentas anteriores.

#### Modifica el password del usuario u3 y desbloquéalo en una única instrucción SQL

#### Da de baja al usuario u1.

### Ampliación

#### Modifica la variable del sistema para hacer que todos los usuarios creados tengan que cambiar el password cada 30 días.

* Crea un nuevo usuario.
* Muestra las opciones empleadas para crear dicho usuario.
* Deja la variable del sistema a su valor original sin utilizar el número 360.

#### Crea tres usuarios de nombre u1,u2,u3 utilizando una única instrucción SQL.

* Haz que cualquier usuario que se cree, pueda por defecto, realizar como máximo 3 conexiones simultaneas al servidor.
* Crea un usuario y comprueba que esto es cierto. Visualiza el valor de la variable del sistema que afecta al usuario (a nivel de sesión)
* Crea dos usuarios utilizando una única orden SQL, de tal forma que las dos cuentas estén inicialmente bloqueadas y que el máximo número de conexiones simultáneas sea de 1.
* Ampliación: Debes obligar a que dichos usuarios utilicen conexiones seguras.

#### Desbloquea las cuentas anteriores.

#### Modifica los usuarios u1 y u2 utilizando una única instrucción SQL y elimina la restricción de que el número máximo de operaciones sea de cincuenta, pero ponles la limitación de que el máximo número de consultas sea de cien a la hora.

#### Modifica el password del usuario u3 y desbloquéalo en una única instrucción SQL

#### Da de baja al usuario u1.

## MySQLWorkBench

#### Crea un usuario de nombre work2 que se pueda conectar desde cualquier red con IP 10.1.1.X.

#### Modifica el usuario anterior y bloquea su cuenta.

#### Borra el usuario anterior.

### Ampliación

#### Crea un usuario de nombre work2 que se pueda conectar desde cualquier red con IP 10.1.1.X.

#### Modifica el usuario anterior e impide que pueda hacer más de 10 operaciones de modificación a la hora.

#### Modifica el usuario anterior y oblígalo a que se conecte con conexiones seguras. ¿Como comprobarías desde Workbench que la conexión es segura?

#### Crea una conexión desde tu equipo real al virtualizado utilizando un túnel SSH, haciendo que en tu equipo (el real) una conexión al puerto 4000 se redirija al puerto 3306 del servidor MySQL.

* Crea un usuario en MySQL que tenga permiso de acceso. ¿Desde qué IP? (la pregunta tiene trampa)
* Crea una conexión en MySQL Workbench conectándote con este usuario.

# GESTIÓN DE PERMISOS

## GRANT

#### Crea un usuario 'tengopermisos' y otórgale permisos para que pueda crear usuarios.

#### Conectado como 'tengopermisos' crea un nuevo usuario 'user1'.

#### Conectado como 'root', otórgale permisos al usuario 'user1' para que pueda parar el servicio MySQL.

#### Conectado como 'root', muestra los permisos que tiene el usuario 'user1'.

Conectado como 'user1' muestra los permisos que posee y comprueba que son los mismos a los de la orden anterior.

#### Conectado como 'root' otorga permiso de creación y borrado de procedimientos, así como de ejecución al usuario 'user1' sobre una base de datos creada previamente.

#### Conectado como 'root' crea un usuario de nombre 'creartablas' que tenga permisos para crear, borrar y modificar tablas de una base de datos creada previamente.

#### Conectado como 'creartablas' crea una tabla sencilla de al menos dos columnas.

#### Conectado como 'root' crea un usuario de nombre 'accesoglobal' que pueda realizar operaciones de selección y inserción sobre todas las tablas de todas las bases de datos.

#### Conectado como 'accesoglobal' añade una fila a la tabla creada anteriormente. Intenta borrar la fila creada. ¿Puedes?

#### Conectado como 'root' crea un usuario de nombre 'accesolocal' que pueda seleccionar todas las tablas de la base de datos anterior.

Conéctate como 'accesolocal' y comprueba que puedes selecciona la fila añadida anteriormente.

#### Conectado como 'root' crea un usuario de nombre 'accesolimitado' que pueda realizar operaciones de inserción, actualización y selección sobre la primera columna de la tabla creada previamente.

Conéctate como 'accesolimitado' y comprueba que tienes los permisos ejecutando las órdenes SQL SELECT, UPDATE e INSERT.

Comprueba que permisos tienes.

### Ampliación

#### Crea un usuario 'tengopermisos' y otórgale permisos para que pueda crear usuarios.

#### Conectado como 'tengopermisos' crea un nuevo usuario 'user1'.

#### Conectado como 'root', otórgale permisos al usuario 'user1' para que pueda crear tablespaces.

Conectado como 'user1' comprueba que dispone de dichos permisos ejecutando sentencias SQL que necesiten tener el permiso otorgado.

#### Conectado como 'root', muestra los permisos que tiene el usuario 'user1'.

Conectado como 'user1' muestra los permisos que posee y comprueba que son los mismos a los de la orden anterior.

#### Conectado como 'root' crea un usuario 'matador' que tenga permisos para poder eliminar conexiones activas y pueda ver las conexiones activas de todos los usuarios con el servidor.

#### Conectado como 'root' otorga permiso de creación y borrado de procedimientos, así como de ejecución al usuario 'user1' sobre una base de datos creada previamente.

Conectado como 'user1' crea un procedimiento almacenado en la base de datos indicada en el paso anterior, con las siguientes órdenes SQL:

DELIMITER $$

**DROP** **PROCEDURE** **IF** **EXISTS** test\_mysql\_while\_loop$$

**CREATE** **PROCEDURE** test\_mysql\_while\_loop**()**

**BEGIN**

**DECLARE** x INT**;**

**SET** x **=** 1**;**

**WHILE** x **>=** 0 **DO**

**set** x **=** x+1**;**

**END** **WHILE;**

**END**$$

DELIMITER **;**

Este procedimiento crea un bucle infinito. Conéctate como 'user1' y ejecuta el procedimiento con la orden SQL: call test\_mysql\_while\_loop

**Nota**: Si te da algún tipo de error, prueba a cerrar la conexión y volver a conectarte.

#### Conectado como 'matador' identifica el proceso y mátalo. ¿Cómo harías para determinar cuál es la consulta que está consumiendo los recursos de MySQL? Hazlo.

#### Conectado como 'root' crea un usuario de nombre 'creartablas' que tenga permisos para crear, borrar y modificar tablas de una base de datos creada previamente.

#### Conectado como 'creartablas' crea una tabla sencilla de al menos dos columnas.

#### Conectado como 'root' crea un usuario de nombre 'accesoglobal' que pueda realizar operaciones de selección e inserción sobre todas las tablas de todas las bases de datos.

#### Conectado como 'accesoglobal' añade una fila a la tabla creada anteriormente. Intenta borrar la fila creada. ¿Puedes?

#### Conectado como 'root' crea un usuario de nombre 'accesolocal' que pueda seleccionar todas las tablas de la base de datos anterior.

Conéctate como 'accesolocal' y comprueba que puedes seleccionar la fila añadida anteriormente.

#### Conectado como 'root' crea un usuario de nombre 'accesolimitado' que pueda realizar operaciones de inserción, actualización y selección sobre la primera columna de la tabla creada previamente.

Conéctate como 'accesolimitado' y comprueba que tienes los permisos ejecutando las órdenes SQL SELECT, UPDATE e INSERT.

Comprueba qué permisos tienes.

## GRANT / REVOKE / WITH GRANT OPTION

#### Conectado como root crea un usuario de nombre 'creador' que tenga permisos para crear usuarios.

#### Conectado como 'creador' crea un nuevo usuario de nombre 'prueba1'.

Conectado como root haz que tenga permisos de selección y borrado a nivel global y todos los permisos sobre una base de datos de ejemplo creada previamente. Dichos permisos podrán ser gestionados por el usuario.

#### Conectado como 'root' crea un usuario de nombre 'prueba2' que tenga permiso para actualizar una tabla de una base de datos creada previamente. Podrá gestionar dicho permiso.

#### Conectado como 'creador' crea un usuario 'prueba3' y 'prueba4'

#### Conectado como 'prueba1' otorga permiso de selección, actualización de una columna de una tabla (creada previamente en la base de datos de ejemplo) y ejecución de procedimientos al usuario 'prueba3' en la base de datos de ejemplo. ¿Puedes hacerlo? ¿Por qué?

#### Conectado como 'root' haz que el usuario 'prueba4' pueda parar el servicio mysql.

#### Impide que el usuario 'prueba1' pueda gestionar los permisos otorgados a nivel global.

Comprueba que aún puede gestionar los permisos de la base de datos.

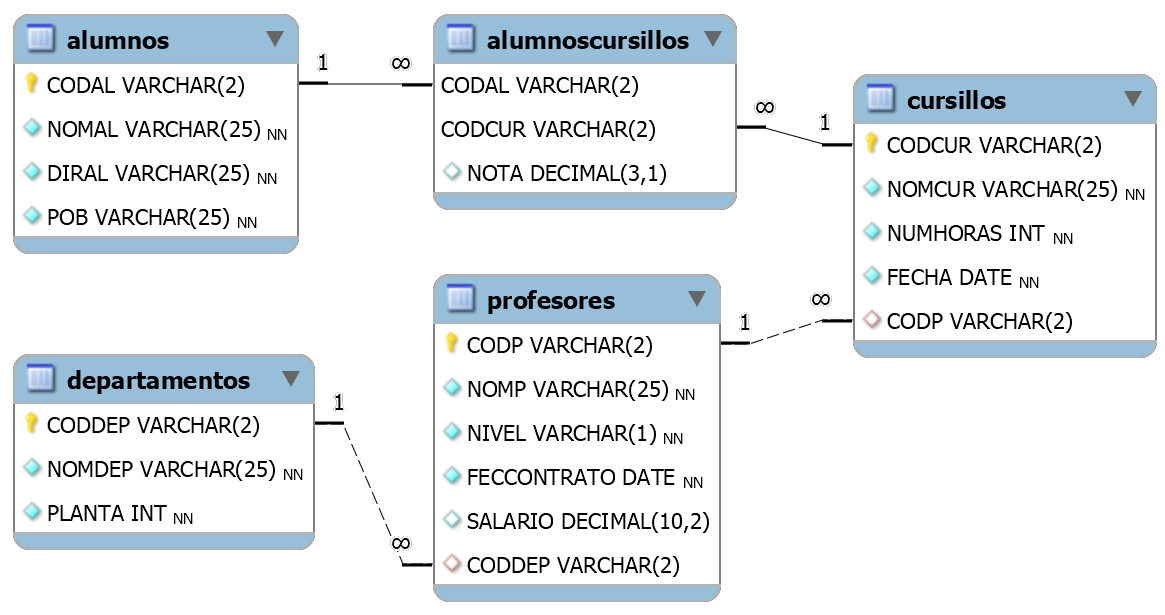
¿Con qué usuario has realizado la operación? ¿Lo puedes hacer conectado con el usuario 'creador'? ¿Por qué?

#### Quita todos los permisos al usuario 'prueba3'. ¿Puedes hacerlo conectado como 'creador'? ¿Por qué?

#### Quita los permisos 'específicos' otorgados a cada uno de los usuarios anteriores, comprobando con la orden SQL SHOW GRANTS que realmente fueron eliminados.

## ROLES

### BD formación



#### Crea los siguientes roles y permisos.

* Administrador: Puede realizar inserciones, borrados, actualizaciones y consultas de datos en todo el servidor. Puede crear, modificar y borrar bases de datos y tablas. Puede crear, modificar y ejecutar rutinas.
* Jefatura\_estudios: Tendrá acceso únicamente a BD formación. Puede realizar inserciones, borrados, actualizaciones y consultas en todas las tablas de BD Formación. Pero no puede borrarlas ni alterarlas.
* Profesor\_titular: Tendrá acceso únicamente a BD formación. Puede realizar consultas en todas las tablas de BD Formación. Pero solo puede hacer inserciones, borrados y actualizaciones en las tablas ALUMNOS Y ALUMNOSCURSILLOS. Por supuesto, no puede borrar ni alterar ninguna tabla.
* Profesor\_apoyo: Tendrá acceso únicamente a BD formación. Puede realizar únicamente consultas en todas las tablas de BD Formación. No tendrá ningún permiso más.

#### Crea 4 usuarios y asígnale a cada uno un rol creado en el ejercicio anterior.

#### Permite a los usuarios con el rol Jefatura\_estudios que puedan otorgar sus permisos a otros usuarios.

#### Autenticado con un usuario con el rol Jefatura\_estudios, dale más permisos a un usuario con el rol Profesor\_Titular y deniégaselos a otro usuario con el rol Profesor\_apoyo.

# AMPLIACIÓN: NIVELES DE SEGURIDAD

#### Muestra utilizando la herramienta cliente 'mysql' el listado de usuarios-hosts del servidor MySQL

#### Crea un usuario gráficamente utilizando la herramienta MySQLWorkBench con nombre 'userWork' y password '1234'. Haz que sólo se pueda conectar desde la red '10.10.1.X'

#### Comprueba la lista de usuarios creados y los permisos que tienen.

#### Crea un usuario anónimo (es decir, que no tenga password) de nombre 'anonino' directamente en la tabla 'user' y que tenga acceso desde cualquier IP.

* Lo puedes hacer con la orden INSERT o gráficamente desde el MySQL WorkBench (debes listar los usuarios presionando el botón derecho sobre la tabla y añadir una fila. Recuerda que debes presionar el botón 'Apply' una vez añadida la fila).
* Al hacerlo directamente vamos a encontrar un problema con 3 columnas que no tienen un valor por defecto y que es necesario asignarles un valor: `ssl\_cipher`,`x509\_issuer`,`x509\_subject`
* La orden SQL para hacerlo sería:

INSERT INTO mysql.user ('Host', 'User', 'Select\_priv', 'ssl\_cipher','x509\_issuer','x509\_subject')

VALUES ('%', 'anonimo', 'Y', '','','');

1. Intenta conectar con dicho usuario (no deberías poder).
2. ¿Qué necesitas hacer para que los nuevos permisos tengan efecto? Hazlo utilizando la orden SQL correspondiente.
3. Comprueba que una vez ejecutada la orden anterior ya puedes conectarte.
4. Una vez conectado, comprueba cuál es tu usuario.
5. Intenta realizar un 'select' sobre una tabla cualquiera. ¿Cuál es el resultado? ¿Por qué?
6. Modifica el usuario anterior (gráficamente o con la orden SQL UPDATE) y haz que tenga permiso de selección sobre todas las tablas de todas las bases de datos.
7. Aplica los permisos sin reiniciar el servidor utilizando la herramienta cliente 'mysqladmin' y comprueba que funciona.
8. Ejecuta la orden SQL que muestre los permisos que tiene concedido el usuario 'anónimo'.

#### Crea un usuario de nombre 'accesotabla' y password 'accesotabla' que tenga acceso desde cualquier ip.

1. Comprueba que no pueda seleccionar la tabla 'alumnos' de la base de datos 'bdFormacion'.
2. Dale permiso de selección sobre la tabla 'alumnos' de la base de datos 'bdFormacion'. Hazlo gráficamente o con la orden sql INSERT. Debes añadir un valor a la columna TimeStamp con el siguiente formato: 2017-09-19 16:41:06
3. Comprueba que puedes seleccionar la tabla. ¿No puedes? ¿Qué te falta por hacer? Hazlo y comprueba que ya puedes seleccionar.

#### Supón que se quieren conectar a tu MySQL todos los equipos del dominio electronica.es pero dentro de este dominio no quieres que se puedan conectar los equipos del dominio **zonanorte.electronica.es**.

* Utilizando lo visto hasta ahora cuales son las entradas que deberás añadir a la tabla 'user', suponiendo que el usuario que se va a conectar siempre es el mismo.
* **Pista**: Fíjate en todas las columnas que conforman la tabla 'user' y busca una que permite impedir el acceso a un usuario.