



1º DAW

U.T. 5

El lenguaje SQL como DCL



Pablo Berciano Posada



Tabla de contenidos

01

**Lenguaje de
control de datos**

02

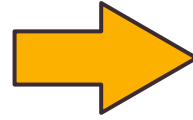
**Usuarios y
límites de uso**

03

**Gestión de
privilegios**

04

**Grupos de
permisos y límites**



01

Lenguaje de control de datos





¿Qué es el DCL?



El **lenguaje de control de datos (DCL)**, por sus siglas en inglés: Data Control Language) es un lenguaje proporcionado por el SGBD que incluye una serie de comandos SQL que permiten al administrador **controlar el acceso a los datos** contenidos en la base de datos. El DCL se utiliza para gestionar usuarios y administrar los permisos y los roles en la base de datos.

Los **comandos DCL** incluyen instrucciones como:

- **GRANT:** Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.
- **REVOKE:** Permite eliminar permisos que previamente se han concedido con GRANT.



Usuarios



Los sistemas de bases de datos **multiusuario** disponen de **mecanismos de seguridad** que controlan el acceso y el uso de la base de datos.

La utilización de usuarios permite definir un **dominio de seguridad**, o lo que es lo mismo, un conjunto de propiedades que determinan aspectos como:

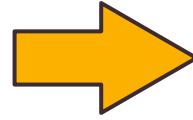
- **Acciones permitidas** al usuario (privilegios y roles)
- **Cuotas** de tablespaces (límites en almacenamiento)
- **Limitación de recursos** del sistema (tiempo de CPU, entre otras)

Usuarios en el catálogo

En MySQL, los usuarios pueden consultarse en una **tabla virtual** llamada **user**, dentro de la base de datos mysql. Con la sentencia **SELECT * FROM mysql.user;** podremos ver tanto los usuarios como el hash de su contraseña y los privilegios de cada uno. También podemos realizar una consulta sobre la tabla **roles_mapping** de mysql para ver los roles creados.

	Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv
<input type="checkbox"/>   	localhost	root		Y	Y	Y	Y
<input type="checkbox"/>   	localhost	test	*A4B6157319038724E3560894F7F932C8886EBFCF	N	N	N	N
<input type="checkbox"/>   	127.0.0.1	root		Y	Y	Y	Y

Por otra parte, la sentencia **SHOW PRIVILEGES;** nos mostrará información sobre los privilegios que podemos asignar a usuarios y roles, mientras que la sentencia **SHOW GRANTS FOR nombre_usuario/rol;** nos mostrará qué privilegios tiene un usuario o rol.



02

Usuarios y límites de uso





Creación y eliminación de usuarios

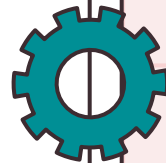
Para crear usuarios que se puedan conectar a nuestra base de datos, utilizaremos el comando:

```
CREATE USER nombre_de_usuario@servidor IDENTIFIED BY 'contraseña';
```

Para acceder a la base de datos como un usuario concreto, modificaremos el comando mysql para que sea algo así:

```
mysql -u nombre_de_usuario -p
```

De esta forma estamos especificando que nos interesa utilizar una contraseña y nos la pedirá a continuación.



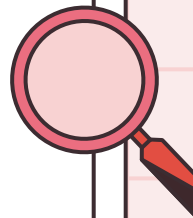


Cuotas de uso

En MySQL existen cuatro opciones de **uso de recursos** que **limitan** a nuestros usuarios. Las **limitaciones evitan abusos de uso** en la BD y **evitan ataques** de denegación de servicio. Los valores establecidos deben ser suficientes para que el usuario pueda trabajar, evitando los abusos y ataques. Las limitaciones disponibles son:

- **MAX_QUERIES_PER_HOUR**: número de consultas que puede realizar en una hora.
- **MAX_UPDATES_PER_HOUR**: número de updates que puede realizar en una hora.
- **MAX_CONNECTIONS_PER_HOUR**: número máximo de conexiones por hora.
- **MAX_USER_CONNECTIONS**: número de conexiones simultáneas al servidor.
- **MAX_STATEMENT_TIME**: número de segundos de ejecución por sentencia.

Todas las anteriores limitaciones pueden **establecerse** a la hora de **creación** de un usuario, aunque también se pueden **cambiar** más adelante. En el caso de no establecer unos valores en la creación del usuario, se tomarán **valores por defecto**.



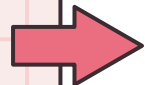


Seguridad

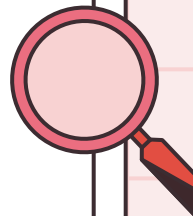


En MySQL, es posible establecer un **máximo número de intentos de conexión**, antes de bloquear la cuenta de un usuario por un tiempo prudente. Esto hace que los ataques de contraseñas de fuerza bruta sean infructuosos. Se haría estableciendo un valor en creación de usuario llamado **FAILED_LOGIN_ATTEMPTS**, junto con otro que especifica el tiempo de bloqueo en días **PASSWORD_LOCK_TIME**.

Por otro lado, es posible establecer una fecha de **caducidad a las contraseñas** de los usuarios, obligándolos así a cambiarla cada cierto tiempo, y se puede guardar una memoria de contraseñas pasadas para que no pongan una repetida. Para establecer caducidad a una contraseña, se usa **PASSWORD EXPIRE INTERVAL días DAY**, y la memoria se establece con **PASSWORD HISTORY num_contraseñas**.



Creación de usuarios con cuotas de uso y limitaciones en la contraseña



Para crear un usuario con limitaciones de uso y contraseña:

```
CREATE OR REPLACE USER user3@localhost IDENTIFIED BY '1234' WITH  
    MAX_USER_CONNECTIONS 5  
    MAX_UPDATES_PER_HOUR 50;
```

Para crear un usuario con caducidad de contraseña y contraseña:

```
CREATE OR REPLACE USER user IDENTIFIED BY password  
    PASSWORD EXPIRE INTERVAL 90 DAY;
```



Modificación de usuarios



Para **modificar un usuario** debemos utilizar la sentencia **ALTER USER**, de esta manera podemos modificar parámetros como limitaciones de uso del usuario y tiempos de caducidad de la contraseña.

Para cambiar la contraseña se haría:

```
ALTER USER user3@localhost IDENTIFIED BY '123';
```

Para añadir una restricción de uso de conexiones:

```
ALTER USER user3@localhost WITH MAX_USER_CONNECTIONS 50;
```

Para retirar la caducidad de una contraseña:

```
ALTER USER user3@localhost PASSWORD EXPIRE NEVER;
```



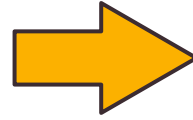
Borrado de usuarios



Para borrar un usuario utilizamos la sentencia:

DROP USER user;

Hay que tener en cuenta que los objetos que haya creado ese usuario pasarán a ser objetos huérfanos, por lo que es importante limpiar primero todas aquellos objetos que haya creado antes de borrar la cuenta.



03

Gestión de privilegios





Privilegios



Un **privilegio** es un **derecho a ejecutar un tipo de sentencia SQL**. Los privilegios de una BD MySQL se pueden separar en tres tipos:

- **Privilegios administrativos:** globales a todo el servidor MySQL
- **Privilegios de BD:** Aplican a una base de datos específica, y todos sus objetos, o sobre todas las BBDD.
- **Privilegios de objetos:** Aplican sobre tablas, índices, vistas, etc. concretas, o sobre un tipo de objeto en general.

Los privilegios se pueden conceder de dos formas distintas:

- **Concesión explícita** de privilegios.
- **Concesión de privilegios a través de roles**, agrupando una serie de privilegios en un rol, al que asignaremos nuestros usuarios del mismo dominio de seguridad.



Privilegios



En MySQL hay muchos tipos de privilegios como:

- **ALL PRIVILEGES:** Concede todos los privilegios disponibles.
- **CREATE [ROLE, USER, etc..]:** Permite crear objetos.
- **DROP:** Permite eliminar bases de datos y tablas.
- **DELETE:** Permite eliminar datos de una tabla.
- **INSERT:** Permite insertar datos en una tabla.
- **SELECT:** Permite leer datos de una tabla.
- **UPDATE:** Permite actualizar datos en una tabla.
- **ALTER:** Permite modificar la estructura de una tabla.
- **REFERENCES:** Permite a un usuario crear y modificar claves externas en tablas.
- **GRANT OPTION:** Permite a un usuario otorgar privilegios a otros usuarios.
- **EXECUTE:** Permite la ejecución de procedimientos y funciones.



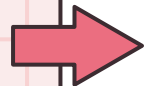
Asignar privilegios a un usuario

Para **asignar privilegios** a un usuario en MySQL hay escribir la sentencia:

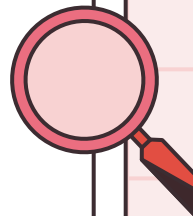
```
GRANT privilege_type[, ...] ON database_name.table_name TO user;
```

Si utilizamos **WITH GRANT OPTION** al final de la sentencia de asignación de privilegios, el usuario en cuestión podrá otorgar sus propios privilegios a otros usuarios, puede ser útil para que el administrador delegue trabajo, pero es potencialmente peligroso. Solo debería otorgarse a usuarios de la suficiente confianza.

Una vez se han concedido permisos, es importante ejecutar **FLUSH PRIVILEGES;** para actualizar los permisos del servidor MySQL.



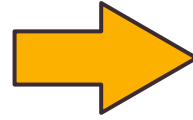
Revocar privilegios a un usuario



Al igual que se pueden otorgar, los privilegios se pueden revocar a los usuario. En MySQL se utiliza la sentencia **REVOKE** para ello:

```
REVOKE privilege_type[, ...] ON database_name.table_name FROM user;
```

Una vez se han revocado los permisos, es importante ejecutar **FLUSH PRIVILEGES**; para actualizar los permisos del servidor MySQL.



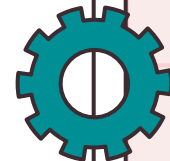
04

Grupos de permisos y límites





Roles



Un **rol** es un **conjunto de privilegios** relacionados por un nombre. Permite conceder varios privilegios de una vez vinculándose primero al rol y luego concediendo este a los usuarios.

MySQL va a tratar los roles como si fueran cuentas de usuarios creados. Tanto es así, que aparecerán en la vista de **mysql.user**, solo diferenciándolos con la columna “**isRole**”. También aparecen los roles en la tabla **roles_mapping** de la base de datos mysql.

Los roles en MySQL son algo **muy reciente**, pudiendo usarse solo a partir de la versión 8.0.

Para **crear un rol** ejecutaremos la siguiente sentencia:

```
CREATE ROLE nombre_rol;
```



Gestión de permisos en los roles

Una vez existen los roles, podemos **darles privilegios** o quitárselos. Para asignar los privilegios de un rol, usaremos la sentencia **GRANT**:

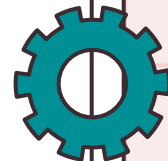
```
GRANT privilegio1 [, ...] ON base_datos.tabla TO nombre_rol;
```

Para revocar los privilegios de un rol, usaremos la sentencia **REVOKE**:

```
REVOKE privilegio1 [, ...] ON base_datos.tabla FROM nombre_rol;
```

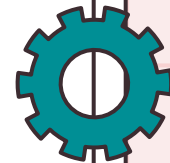
Una vez los roles tienen privilegios, es posible **asignarlos** a los usuarios con la sentencia:

```
GRANT nombre_rol TO nombre_usuario;
```





Perfiles



En MySQL, un **perfil** es un **conjunto de parámetros de configuración** que se pueden **asignar a un usuario** o a una conexión de usuario para controlar su comportamiento. Los perfiles se utilizan para **establecer límites** en los recursos que un usuario puede utilizar. Los perfiles pueden ser muy útiles para **administrar grandes grupos de usuarios** y garantizar que no utilicen más recursos de los que se les ha asignado.

Al crear un perfil en MySQL, se pueden **establecer diversas restricciones**, como límites de tiempo de espera, límites de consultas, límites de conexiones, límites de memoria, entre otros. Estos límites se aplicarán a **todas las consultas** realizadas por los **usuarios** que estén **asignados a ese perfil**.

Los perfiles no se pueden utilizar en la versión comunitaria de MySQL, presente en **XAMPP**.