

Roles y usuarios en MariaDB

Autor: Ambrosio Cardoso Jiménez
Fecha: 06-Septiembre-2020
Objetivo: Crear **roles y usuarios** en MariaDB y asignar permisos

Requisitos:

MariaDB 10.4.13 o superior

El tema de crear usuarios, roles, asignación y revocación de permisos es un tema recurrente en el área de base de datos. Comencemos estableciendo una diferencia entre rol y usuario.

El **rol** es una función que realiza un **usuario** en particular o un grupo de usuarios. Por ejemplo, tengo un conjunto de 6 usuarios en una base de datos de control escolar: Rosa, Angela, Margarita, Roselia, Rubén y José. Rosa y Angela son las secretarias que realizan un conjunto de operaciones definidas en la base de datos como agregar nuevos alumnos, generar lista de calificaciones, crear o modificar grupos; Margarita y José son docentes de la escuela y dentro de la base de datos las operaciones a las que tienen acceso es la de subir calificaciones y realizar cambios en ello, pero no pueden agregar alumnos; por otra parte Roselia y Rubén son los programadores de la Aplicación Web desde donde se administra toda la información y por tanto tiene acceso a la base de datos. Finalmente Roselia es la administradora de la Base de Datos y tiene control total sobre ello. Entonces un **usuario** es la representación de una persona física y el **rol es la función** que realiza dentro de la base de datos. Así podemos concluir en este escenario que **docente, secretaria, programadores y administradora de base de datos** son **roles**, mientras que Rosa, Angela, Margarita, Roselia, Rubén y José son los **usuarios**.

Ahora bien ¿para que se requieren los roles?. Los roles facilitan la concesión o revocación de privilegios lo que significa que se puede asignar los permisos en una sola ocasión en lugar de 10 o 20 usuarios uno a uno. Además muchos usuarios pueden tener privilegios cruzados lo que dificulta la asignación de uno a uno.

Para crear roles se usa la siguiente sintaxis

```
CREATE ROLE [IF NOT EXISTS] role [, role ] ...
```

Ejemplo:

```
CREATE ROLE 'dba', 'developer';
```

La sentencia anterior crea dos roles dba (database administrator) y developer.

Crear los siguientes roles: **docente, secretaria y alumno**

```
CREATE ROLE docente, secretaria, alumno;
```

Para borrar los roles se usa la instrucción **DROP ROLE** *role* [, *role*] ...

```
DROP ROLE dba;
```

Conceder privilegios a los roles

```
CREATE ROLE IF NOT EXISTS developer;
```

```
CREATE ROLE IF NOT EXISTS estudiante;
```

```
CREATE ROLE IF NOT EXISTS secretaria;
```

```
GRANT ALL ON prueba.* TO developer;
```

```
GRANT SELECT ON prueba.calificaciones TO estudiante;
```

```
GRANT INSERT, UPDATE ON prueba.calificaciones TO secretaria;
```

Permisos

Aquí un listado permisos que podemos asignar.

<i>ALL [PRIVILEGES]</i>	permite conceder todos los privilegios (SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE_TMP_TABLE, LOCK_TABLES, CREATE_VIEW, SHOW_VIEW, CREATE_ROUTINE, ALTER_ROUTINE, EXECUTE y GRANT) excepto GRANT OPTION
USAGE	No otorga ningún privilegio
ALTER	Privilegio para alterar la estructura de una tabla
CREATE	Permite el uso de create table
DELETE	Permite el uso de delete
DROP	Permite el uso de drop table
INDEX	Permite el uso de index y drop index
INSERT	Permite el uso de insert

SELECT	Permite el uso de select
UPDATE	Permite el uso de update
FILE	Permite el uso de select . . . into outfile y load data infile
PROCESS	Permite el uso de show full process list
SUPER	Permite la ejecución de comandos de supervisión
RELOAD	Permite el uso de flush
REPLICATION CLIENT	Permite preguntar la localización de maestro y esclavo
REPLICATION SLAVE	Permite leer los binlog del maestro
GRANT OPTION	Permite el uso de grant y revoke
SHUTDOWN	Permite dar de baja al servidor
LOCK TABLES	Permite el uso de lock tables
SHOW TABLES	Permite el uso de show tables
CREATE TEMPORARY TABLES	Permite el uso de create temporary table

En la sentencia **GRANT ALL ON prueba.* TO developer**; lo que estamos indicando es conceder todos los privilegios sobre la base de datos **prueba** y a todas las **tablas (*)** que están en ella al rol **developer**; mientras que en este caso **GRANT SELECT ON prueba.calificaciones TO estudiante**; solo estamos concediendo permiso a la tabla **calificaciones** de la base de datos **prueba** el permiso de solo lectura al rol **estudiante**.

Usuarios.

Para crear un usuario se usa la sentencia CREATE USER, modificar un usuario ALTER USER y borrar DROP USER

```
GRANT SHOW DATABASES ON *.* TO developer;
```

```
CREATE USER ambrosio@'%';
```

```
GRANT developer to ambrosio; -- Asigna usuario ambrosio al rol developer
```

```
ALTER USER ambrosio@'%' IDENTIFIED BY 'seguridadmaxima'; -- Asigna password
```

```
FLUSH PRIVILEGES;
```

Esta última sentencia permite actualizar la caché de los privilegios, es decir vuelve a reconstruir los permisos que están en memoria haciendo que los cambios sean visibles inmediatamente este tiene sentido en una base de datos distribuida. Mientras no se modifique la tabla de privilegios manualmente se uso no debería ser necesario.

Podemos bloquear un rol o usuario utilizando la siguiente sentencia:

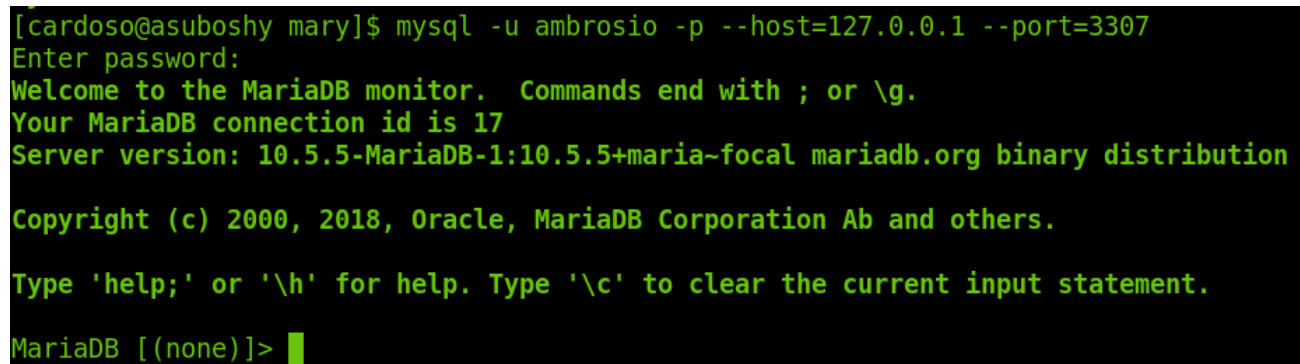
ALTER USER `developer` ACCOUNT LOCK;

o desbloquear

ALTER USER `developer` ACCOUNT UNLOCK;

Caso práctico

Iniciar una **nueva sesión** con la cuenta de **ambrosio** creado anteriormente



```
[cardoso@asuboshy mary]$ mysql -u ambrosio -p --host=127.0.0.1 --port=3307
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.5.5-MariaDB-1:10.5.5+maria~focal mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> █
```

Figura 1. inicio de sesión

Abrir la base de datos con la sentencia `use prueba` (se asume que hay una base de datos llamada `prueba` con una tabla de calificaciones)

MariaDB [(none)]> **use prueba;**

ERROR 1044 (42000): Access denied for user 'ambrosio'@'%' to database 'prueba'

MariaDB [(none)]>

En este caso genera una excepción diciendo que `ambrosio` no tiene privilegios de abrir la base de datos `prueba`. Para ello es necesario ejecutar la siguiente instrucción;

MariaDB [(none)]> **SET ROLE `developer`;**

y volver a abrir la base de datos

MariaDB [(none)]> **use prueba;**

```

MariaDB [(none)]> use prueba;
ERROR 1044 (42000): Access denied for user 'ambrosio'@'%' to database 'prueba'
MariaDB [(none)]> SET ROLE developer;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> use prueba;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [prueba]> █

```

Figura 2. Privilegio heredado del rol

La sentencia **SET ROLE developer**; permite habilitar los privilegios para los usuarios que tienen asignado el rol de **developer**.

También podemos limitar el consumo de recursos por ejemplo:

```

GRANT ALL PRIVILEGES ON prueba.* TO 'ambrosio'@'%' IDENTIFIED BY 'seguridadmaxima'
WITH MAX_CONNECTIONS_PER_HOUR 2 MAX_QUERIES_PER_HOUR 200
MAX_UPDATES_PER_HOUR 50;

```

MAX_CONNECTIONS_PER_HOUR → Número de conexiones por hora.

MAX_QUERIES_PER_HOUR → Número de conexiones por hora.

MAX_UPDATES_PER_HOUR → Número de actualizaciones por hora.

Para probar este permiso abrimos tres consolas e iniciamos sesión con la cuenta de **ambrosio**

```

[cardoso@asuboshy mary]$ mysql -u ambrosio -p --host=127.0.0.1 --port=3307
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.5.5-MariaDB-1:10.5.5+maria~focal mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █

```

Figura 3. Consola 1

```
[cardoso@asuboshy mary]$ mysql -u ambrosio -p --host=127.0.0.1 --port=3307
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.5.5-MariaDB-1:10.5.5+maria-focal mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figura 4. Consola 2

```
[cardoso@asuboshy mary]$ mysql -u ambrosio -p --host=127.0.0.1 --port=3307
Enter password:
ERROR 1226 (42000): User 'ambrosio' has exceeded the 'max_connections_per_hour' resource (current value: 2)
[cardoso@asuboshy mary]$ █
```

Figura 5. Consola 3

En la figura 5 se aprecia el mensaje de que el usuario **ambrosio** está excediendo del número de conexiones permitidos.

NIVELES DE PRIVILEGIOS

Globales: Se aplican al conjunto de todas las bases de datos en un servidor. Es el nivel más alto de privilegio, en el sentido de que su ámbito es el más general.

- Estos privilegios son almacenados en la tabla **mysql.user**
- **GRANT ALL ON *.*** y **REVOKE ALL ON *.*** otorgan y quitan sólo permisos globales.

De base de datos: Se refieren a bases de datos individuales, y por extensión, a todos los objetos que contiene cada base de datos.

- Estos permisos se almacenan en las tablas **mysql.db**
- **GRANT ALL ON db_name.*** y **REVOKE ALL ON db_name.*** otorgan y quitan sólo permisos de bases de datos.

De tabla: Se aplican a tablas individuales, y por lo tanto, a todas las columnas de esas tabla.

- Estos permisos se almacenan en la tabla **mysql.tables_priv**
- GRANT ALL ON db_name. tbl_name y REVOKE ALL ON db_name. tbl_name otorgan y quitan permisos sólo de tabla.

De columna: Se aplican a una columna en una tabla concreta.

- Estos permisos se almacenan en la tabla **mysql.columns_priv**
- Usando REVOKE, debe especificar las mismas columnas que se otorgaron los permisos

```
CREATE USER juanito IDENTIFIED BY 'pwdJuanito';  
GRANT EXECUTE ON prueba.* to juanito@'%';  
GRANT SELECT (estudiante_id, materia_id, docente_id, calif1)  
ON prueba.calificaciones to juanito@'%';
```

MariaDB [prueba]> SELECT * FROM calificaciones;

ERROR 1142 (42000): SELECT command denied to user 'juanito'@'172.17.0.1' for table 'calificaciones'

NO TIENE PRIVILEGIOS PARA VER TODAS LAS COLUMNAS (*) DE LA TABLA calificaciones

De rutina: Se aplican a los procedimientos almacenados y funciones.

GRANT EXECUTE ON FUNCTION | PROCEDURE nombreFuncion o nombreProcedimiento TO usuario

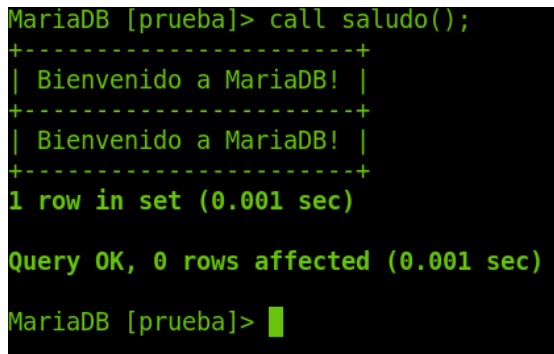
Ejemplo: Ejecutar este bloque con cuenta de administrador (root)

```
DELIMITER //  
  
CREATE OR REPLACE PROCEDURE saludo ()  
  
BEGIN  
  
    SELECT 'Bienvenido a MariaDB!';  
  
END;  
  
//  
  
DELIMITER ;
```

GRANT EXECUTE ON PROCEDURE saludo TO juanito@'%';

Desde la terminal dónde está conectado juanito ejecutar

CALL saludo();



```
MariaDB [prueba]> call saludo();
+-----+
| Bienvenido a MariaDB! |
+-----+
| Bienvenido a MariaDB! |
+-----+
1 row in set (0.001 sec)

Query OK, 0 rows affected (0.001 sec)

MariaDB [prueba]> █
```

Figura 6. Ejecución de procedimiento con permiso concedido

CÓMO SE ACCEDE A LA BASE DE DATOS

Nivel 1. Comprobación de la conexión

- Desde dónde se conecta el usuario
- Nombre del usuario
- Consulta a tabla **mysql.user** (host, user, password)

Nivel 2. Comprobación de privilegios

- Por cada petición en la conexión se comprueba si hay privilegios para efectuarla
- Consulta a tablas user, db, tables_priv, columns_priv, procs_priv

Método abreviado para crear una cuenta de usuario y asignar privilegios a la vez

GRANT ALL ON *.* TO usuarioX@'%' IDENTIFIED BY 'pwdDelNuevoUsuario';

Esta instrucción crea el usuario usuarioX con posibilidad de conectarse desde cualquier máquina con una contraseña y privilegios **ALL(vea página anterior)** para todas las bases de datos.

Autenticación vía unix_socket (Sólo para usuarios linux)

GRANT ALL PRIVILEGES ON *.* TO 'cardoso'@'localhost' IDENTIFIED **VIA unix_socket**;

Para utilizar autenticación con cifrado de curva elíptica se debe instalar el siguiente plugin desde la consola de mariadb con cuenta root

```
SQL> INSTALL SONAME 'auth_ed25519';
```

y además se puede indicar en el archivo de configuración que sea el método predeterminado de cifrado editando el archivo `/etc/my.cnf.d/mariadb-server.cnf`

...

```
[mariadb]
```

```
#--- agregado por cardoso
```

```
plugin_load_add = auth_ed25519
```

...

```
SQL> CREATE USER rosa@'%' IDENTIFIED VIA ed25519 USING PASSWORD('hermosaflor');
```

o se puede crear el usuario y conceder privilegios a la vez

```
GRANT SELECT ON test.* TO lilia@'%' IDENTIFIED VIA ed25519 USING  
PASSWORD('aromadulce');
```

```
MariaDB [(none)]> select user, host, password, plugin from mysql.user;
```

User	Host	Password	plugin
mariadb.sys	localhost		mysql_native_password
root	localhost	*9350753A9C9FE445B81C4B00621411872F839C74	mysql_native_password
mysql	localhost	invalid	mysql_native_password
cardoso	localhost		unix_socket
rosa	%		ed25519
lilia	%		ed25519

```
6 rows in set (0.002 sec)  
  
MariaDB [(none)]> █
```

Figura 6. Métodos de cifrado