

# MySQL

## Docs\_

### MySQL: Crear un nuevo usuario y otorgar permisos



# MySQL: Crear un nuevo usuario y otorgar permisos

## Introducción

MySQL es un software de administración de bases de datos de código abierto que ayuda a los usuarios a almacenar, organizar y recuperar datos. Tiene diversas opciones para otorgar permisos matizados a usuarios específicos en las tablas y bases de datos. Este tutorial dará una breve descripción general de algunas de esas opciones.

## Cómo crear un nuevo usuario

Comencemos creando un nuevo usuario en el shell de MySQL:

```
1. CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

Nota: Al añadir usuarios en el shell de MySQL en este tutorial, especificaremos el host del usuario como localhost, y no la dirección IP del servidor. localhost es un nombre de host que significa “este equipo”, y MySQL trata este nombre de host en particular de manera especial: cuando un usuario con ese host inicia sesión en MySQL, intentará conectarse al servidor local utilizando un archivo de socket de Unix. Por lo tanto, localhost se utiliza normalmente cuando planea conectarse implementando SSH a su servidor o cuando está ejecutando el cliente mysql local para conectarse al servidor MySQL local.

En este momento, newuser no tiene permisos para hacer nada con las bases de datos. De hecho, incluso si newuser intenta iniciar sesión (con la contraseña, password), no podrá acceder al shell de MySQL.

Por lo tanto, lo primero que se debe hacer es proporcionar al usuario acceso a la información que necesitará.

```
1. GRANT ALL PRIVILEGES ON *.* TO 'newuser'@'localhost';
```

Los asteriscos en este comando se refieren a la base de datos y la tabla (respectivamente) a los que pueden acceder. Este comando específico permite al usuario leer, editar, ejecutar y realizar todas las tareas en todas las bases de datos y tablas.

Tenga en cuenta que en este ejemplo estamos otorgando a newuser acceso de root completo a todo en nuestra base de datos. Si bien esto es útil para explicar algunos conceptos de MySQL, puede ser poco práctico para la mayoría de casos de uso y podría poner la seguridad de su base de datos en alto riesgo.

# MySQL: Crear un nuevo usuario y otorgar permisos

Una vez que haya finalizado los permisos que desea configurar para sus nuevos usuarios, asegúrese siempre de volver a cargar todos los privilegios.

```
1. FLUSH PRIVILEGES;
```

Sus cambios ahora estarán vigentes.

## Cómo otorgar diferentes permisos de usuario

Aquí se incluye una breve lista de otros posibles permisos comunes que los usuarios pueden disfrutar.

- **ALL PRIVILEGES:** Como vimos antes, esto le otorgaría a un usuario de MySQL acceso completo a una base de datos designada (o si no se selecciona ninguna base de datos, acceso global a todo el sistema).
- **CREATE:** Permite crear nuevas tablas o bases de datos.
- **DROP:** Permite eliminar tablas o bases de datos.
- **DELETE:** Permite eliminar filas de las tablas.
- **INSERT:** Permite insertar filas en las tablas.
- **SELECT:** Les permite usar el comando `SELECT` para leer las bases de datos.
- **UPDATE:** Permite actualizar las filas de las tablas.
- **GRANT OPTION:** Permite otorgar o eliminar privilegios de otros usuarios.

Para proporcionar un permiso a un usuario específico, puede usar este marco:

```
1. GRANT type_of_permission ON database_name.table_name TO 'username'@'localhost';
```

Si desea darle a un usuario acceso a cualquier base de datos o a cualquier tabla, asegúrese de poner un asterisco (\*) en el lugar del nombre de la base de datos o de la tabla.

Cada vez que actualice o cambie un permiso, asegúrese de usar el comando Flush Privileges.

Si necesita revocar un permiso, la estructura es casi la misma que para otorgar un permiso:

```
1. REVOKE type_of_permission ON database_name.table_name FROM 'username'@'localhost';
```

Tenga en cuenta que, cuando revoca permisos, la sintaxis requiere que utilice `FROM` en lugar de `TO`, como se utiliza para otorgar permisos.

# MySQL: Crear un nuevo usuario y otorgar permisos

Puede revisar los permisos actuales de un usuario ejecutando lo siguiente:

```
1. SHOW GRANTS FOR 'username'@'localhost';
```

Al igual que puede eliminar bases de datos con DROP, también puede usar DROP para eliminar un usuario por completo:

```
1. DROP USER 'username'@'localhost';
```

Para probar su nuevo usuario, cierre sesión escribiendo lo siguiente:

```
1. quit
```

y vuelva a iniciar sesión con este comando en el terminal:

```
1. mysql -u [username] -p
```

# MySQL: Crear un nuevo usuario y otorgar permisos

## Ejemplos

### Crear un nuevo usuario y otorgarle privilegios para acceder a una base de datos:

```
CREATE USER 'nuevo_usuario'@'localhost' IDENTIFIED BY 'contraseña';  
GRANT ALL PRIVILEGES ON basedatos.* TO 'nuevo_usuario'@'localhost';  
FLUSH PRIVILEGES;
```

En este ejemplo, se crea un nuevo usuario llamado "nuevo\_usuario" con una contraseña especificada. Luego, se otorgan todos los privilegios para acceder a la base de datos "basedatos". La instrucción FLUSH PRIVILEGES actualiza los privilegios para que el usuario recién creado tenga acceso inmediato.

### Eliminar un usuario existente:

```
DROP USER 'usuario_existente'@'localhost';
```

Este ejemplo elimina un usuario existente llamado "usuario\_existente" con acceso local.

### Cambiar la contraseña de un usuario existente:

```
SET PASSWORD FOR 'usuario_existente'@'localhost' = 'nueva_contraseña';
```

En este ejemplo, se cambia la contraseña del usuario "usuario\_existente" a "nueva\_contraseña".

### Ver los usuarios existentes y sus permisos:

```
SELECT user, host, authentication_string FROM mysql.user;
```

Esta consulta devuelve una lista de todos los usuarios existentes y su información de autenticación.

### Conceder todos los permisos para una base de datos específica:

```
GRANT ALL PRIVILEGES ON basedatos.* TO 'nombre_usuario'@'localhost';
```

En este ejemplo, el usuario "nombre\_usuario" obtiene todos los permisos para acceder a la base de datos "basedatos" desde la máquina local.

# MySQL: Crear un nuevo usuario y otorgar permisos

## Conceder permisos específicos para una tabla:

```
GRANT SELECT, INSERT, UPDATE ON basedatos.tabla1 TO 'nombre_usuario'@'localhost';
```

En este ejemplo, el usuario "nombre\_usuario" obtiene permisos SELECT, INSERT y UPDATE para la tabla "tabla1" en la base de datos "basedatos".

## Conceder permisos para todas las bases de datos:

```
GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'localhost';
```

En este ejemplo, el usuario "nombre\_usuario" obtiene todos los permisos para todas las bases de datos en el servidor MySQL.

## Revocar permisos específicos para una tabla:

```
REVOKE SELECT, INSERT ON basedatos.tabla1 FROM 'nombre_usuario'@'localhost';
```

En este ejemplo, el usuario "nombre\_usuario" pierde los permisos SELECT e INSERT para la tabla "tabla1" en la base de datos "basedatos".

Recuerda que para aplicar los cambios de permisos, se debe ejecutar la instrucción `FLUSH PRIVILEGES` después de la concesión o revocación de permisos.