



Nombre de la práctica	Práctica 3: Roles de usuario			No.	1
Asignatura:	Taller de Base de datos	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: Vanesa Hernández Martínez

GRUPO: 3501

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Actividades en aula de clases y en equipo personal

III. Material empleado:

- Laptop
- Navicat
- Docker

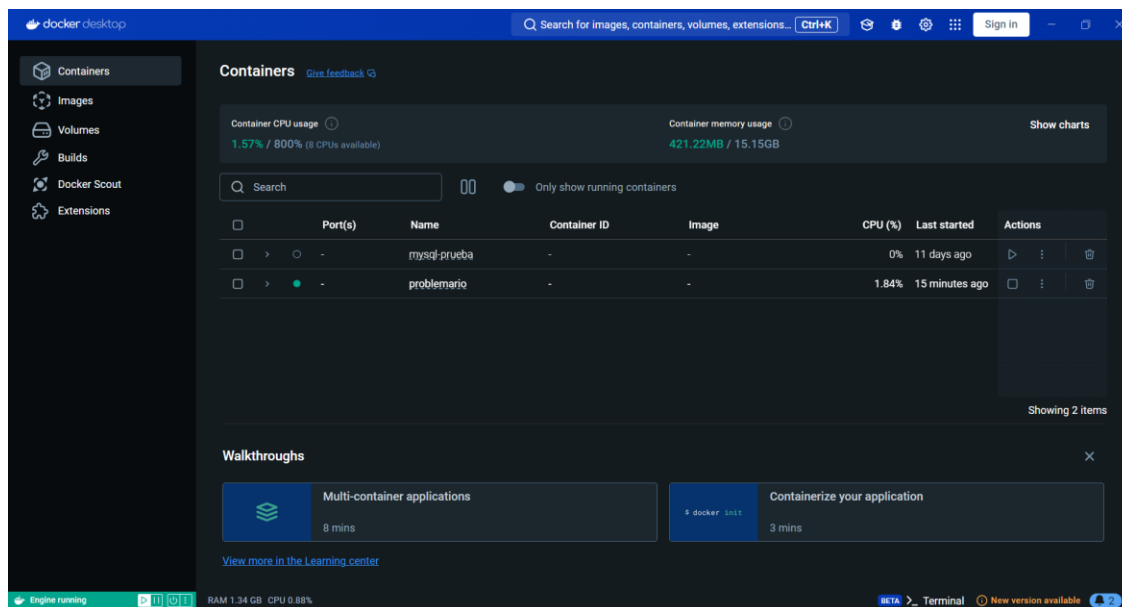
1. Para generar los problemas del tema creamos un nuevo contenedor, para ello creamos un archivo Docker-compose.yaml en donde configuramos todos los datos de la conexión:

```

docker-compose.yaml X
docker-compose.yaml
1  services:
2    database:
3      image: mysql:8.0
4      ports:
5        - 3308:3306
6      volumes:
7        - ./mysql:/var/lib/mysql
8      environment:
9        MYSQL_ROOT_PASSWORD: C413b # Cambia esta contraseña por una segura
10       MYSQL_DATABASE: problemario # Nombre de la base de datos inicial
11       MYSQL_USER: root # Usuario inicial de MySQL
12       MYSQL_PASSWORD: C413b
13
14
15     flask:
16       build: ./app
17       ports:
18         - 5000:5000
19       depends_on:
20         - database

```

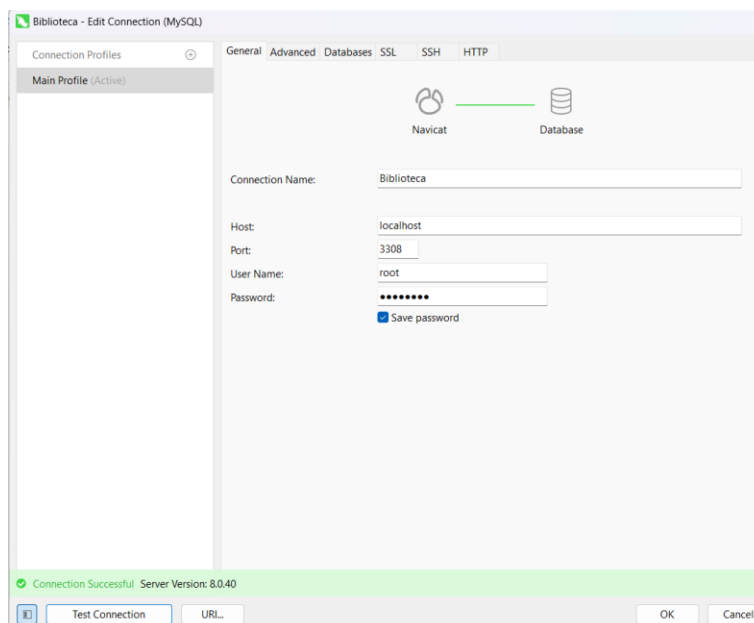
2. Abrimos Docker para revisar que el servicio este corriendo.



3. Dentro de la terminal de Visual colocamos la siguiente instrucción para descargar la imagen y hacer posible la conexión.

docker compose up --build -d

4. Una vez se haya inicializado todo de manera correcta creamos una nueva conexión en Navicat con los datos que especificamos en Docker-compose.yaml, si al hacer testing esta nos muestra una línea de color verde esto indica que la conexión se hizo de manera correcta.



Ejercicio 1: Crear roles básicos

Crea dos roles:

1. rol_lector : Permite solo lectura (SELECT) en la base de datos biblioteca .
2. rol_editor : Permite modificar datos (INSERT , UPDATE , DELETE) en la tabla libros .

1. Código

CREATE ROLE 'rol_lector';

GRANT SELECT ON biblioteca.* TO 'rol_lector';

Explicación:

- ✓ CREATE ROLE 'rol_lector': Crea un nuevo rol llamado rol_lector.
- ✓ GRANT SELECT ON biblioteca.* TO 'rol_lector': Asigna al rol rol_lector el permiso de SELECT en todas las tablas de la base de datos biblioteca, lo que permite solo lectura en esa base de datos.

CREATE ROLE 'rol_editor';

GRANT INSERT, UPDATE, DELETE ON biblioteca.libros TO 'rol_editor';

Explicación:

- ✓ CREATE ROLE 'rol_editor': Crea un nuevo rol llamado rol_editor.
- ✓ GRANT INSERT, UPDATE, DELETE ON biblioteca.libros TO 'rol_editor': Asigna al rol rol_editor los permisos de INSERT, UPDATE y DELETE en la tabla libros de la base de datos biblioteca, lo que permite modificar los datos en esa tabla.

✓ 2. Funcionamiento

SHOW GRANTS FOR 'rol_lector'@'%';

SHOW GRANTS FOR 'rol_editor'@'%';

```
Grants for rol_lector@%  
GRANT USAGE ON *.* TO `rol_lector`@`%`  
GRANT SELECT ON `biblioteca`.* TO `rol_lector`@`%`
```

Ejercicio 2: Asignar roles a usuarios

Asigna el rol `rol_lector` al usuario `lector` y el rol `rol_editor` al usuario `editor`.

1. Código

```
GRANT 'rol_lector' TO 'lector'@'%';
```

Este comando asigna el rol `rol_lector` al usuario `lector`, lo que le permitirá tener permisos de solo lectura en la base de datos `biblioteca`.

```
GRANT 'rol_editor' TO 'editor'@'%';
```

Este comando asigna el rol **rol_editor** al usuario **editor**, otorgándole permisos para insertar, actualizar y eliminar datos en la tabla **libros** de la base de datos **biblioteca**.

✓ 2. Funcionamiento

```
SHOW GRANTS FOR 'lector'@'%';
```

```
SHOW GRANTS FOR 'editor'@'%';
```

```
Grants for lector@%  
GRANT SHOW DATABASES ON *.* TO `lector`@`%`  
GRANT SELECT ON `biblioteca`.* TO `lector`@`%`  
GRANT `rol_lector`@`%` TO `lector`@`%`
```

Ejercicio 3: Revocar un rol

El usuario editor ya no debe tener permisos de edición. Revoca el rol asignado

1. Código

REVOKE 'rol_editor' FROM 'editor'@'%';

- ✓ **REVOKE:** Es el comando utilizado para quitar o revocar permisos o roles que han sido previamente otorgados a un usuario.
- ✓ **'rol_editor':** Es el nombre del **rol** que se está revocando. En este caso, el rol **rol_editor** es el que le da permisos de edición (como **INSERT**, **UPDATE** y **DELETE**) sobre la tabla **libros** en la base de datos **biblioteca**.
- ✓ **FROM:** Es la palabra clave que indica que vamos a quitar un permiso o rol de un usuario.
- ✓ **'editor':** Es el **nombre de usuario** al que se le va a revocar el rol. En este caso, el usuario **editor** es quien pierde el acceso a los permisos de edición.
- ✓ **@'%':** El **@'%'** especifica el **host** desde el cual el usuario puede conectarse a la base de datos. El **'%'** indica que el usuario **editor** puede conectarse desde cualquier host.

2. Funcionamiento

SHOW GRANTS FOR 'editor'@'%';

Grants for editor@%

GRANT USAGE ON *.* TO `editor`@`%`

GRANT INSERT ON `biblioteca`.`libros` TO `editor`@`%`

Ejercicio 4: Activar roles

Los usuarios no pueden usar un rol asignado hasta que lo activen explícitamente. Activa el rol `rol_lector` para el usuario `lector` en su sesión actual.

1. Código

SET ROLE 'rol_lector';

- ✓ **SET ROLE:** Este comando se usa para activar uno o varios roles en la sesión actual del usuario.
- ✓ **'rol_lector':** Es el nombre del rol que se desea activar.

2. Funcionamiento

SHOW GRANTS FOR 'root'@'%';

Grants for root@%
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUT
GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ABORT_EXEMPT,AUDIT.
GRANT `rol_lector`@`%` TO `root`@`%`

Ejercicio 5: Ver roles asignados

Consulta los roles asignados a los usuarios para verificar configuraciones

1. Código

```
SELECT * FROM information_schema.role_table_grants WHERE grantee =  
        "'usuario'@'host'";
```

Este comando devuelve los roles que han sido asignados al usuario especificado.

'usuario': Reemplaza esto con el nombre del usuario.

'host': Reemplázalo con el nombre del host o % si el usuario puede conectarse desde cualquier host.

✓ 2. Funcionamiento

GRANTOR	GRANTOR_HOST	GRANTEE	GRANTEE_HOST	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	PRIVILEGE_TYPE	IS_GRANTABLE
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

Ejercicio 6: Crear un rol combinado

Crea un rol llamado rol_administrador que combine los permisos de rol_lector y rol_editor .
Luego, asígnalo al usuario admin_biblioteca .

1. Código

- ✓ Crear el rol combinado: CREATE ROLE 'rol_administrador';
- ✓ Otorgar los permisos de los roles existentes:
- ✓ GRANT 'rol_lector' TO 'rol_administrador';
- ✓ GRANT 'rol_editor' TO 'rol_administrador';
- ✓ Asignar el rol al usuario: GRANT 'rol_administrador' TO 'admin_biblioteca'@'%';
- ✓ Activar el rol en la sesión del usuario: SET ROLE 'rol_administrador';

2. Funcionamiento

SHOW GRANTS FOR 'root'@'%';

Grants for root@%

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUT

GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ABORT_EXEMPT,AUDIT

GRANT `rol_administrador`@`%`,`rol_lector`@`%` TO `root`@`%`



Conclusión

Los roles de usuario en MySQL facilitan la gestión de permisos al permitir agrupar permisos relacionados y asignarlos a los usuarios de manera eficiente. Esto mejora la seguridad, ya que se pueden definir claramente qué operaciones puede realizar cada usuario.

- **Ventajas:** Simplifica la administración, mejora la seguridad y es flexible para adaptarse a diferentes necesidades.
- **Comandos clave:** CREATE ROLE, GRANT, SET ROLE y REVOKE son los comandos principales para gestionar roles y asignar permisos a los usuarios.