



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Tarea 3

Bases de Datos (1644)

Profesor: Ing. Fernando Arreola Franco

Semestre 2026-2

Grupo: 1

Alumna: Cruz Basilio Ximena Carolina

No. de cuenta: 321116424

Fecha de entrega: 18 de febrero de 2026

I. ADMINISTRACIÓN DE USUARIOS Y ROLES EN POSTGRESQL

I-A. Introducción

En esta práctica trabajé con PostgreSQL utilizando pgAdmin 4 como interfaz gráfica. El objetivo fue aprender a crear usuarios, asignarles restricciones de seguridad y posteriormente crear roles con permisos específicos sobre una tabla. Estas actividades permiten entender cómo funciona el control de acceso dentro de un sistema gestor de bases de datos y cómo se protege la información.

Durante la práctica utilicé la herramienta *Query Tool*, que funciona como una terminal SQL donde se ejecutan los comandos directamente en el servidor de PostgreSQL.

I-B. Entorno de trabajo

Abrí pgAdmin 4 y me conecté al servidor local PostgreSQL 18. Dentro del panel izquierdo seleccioné la base de datos `postgres` y abrí la terminal SQL para comenzar a ejecutar instrucciones.

II. ACTIVIDAD 1: CREAR UN USUARIO CON CONTRASEÑA, LÍMITE DE CONEXIONES Y VIGENCIA

El primer ejercicio consistió en crear un usuario que tuviera:

- Contraseña
- Límite de conexiones
- Vigencia de 1 mes

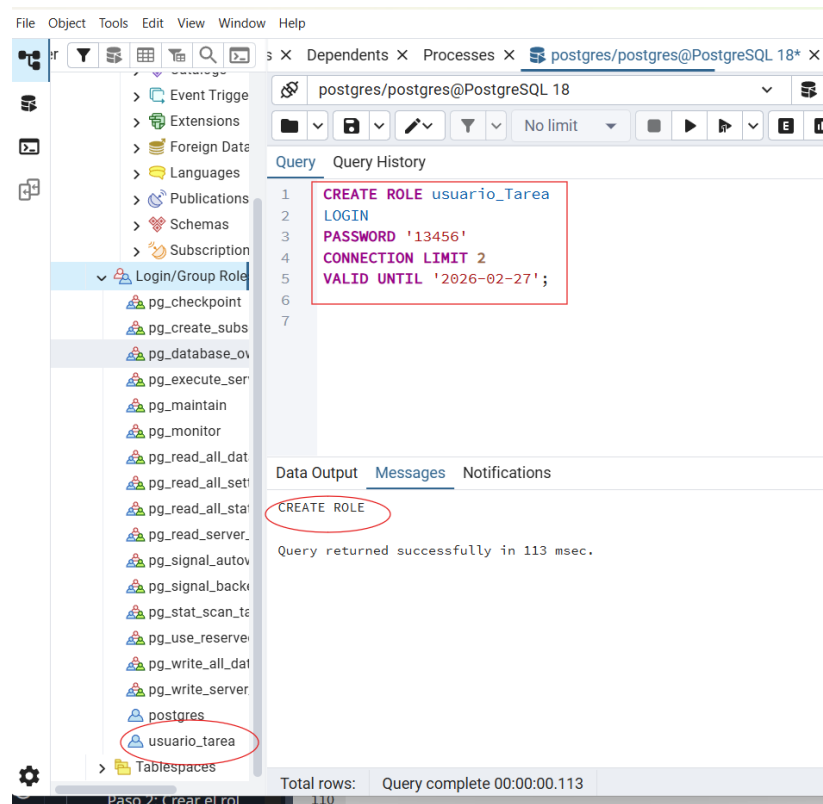
Para ello utilicé el siguiente comando:

```
CREATE ROLE usuario_Tarea
LOGIN
PASSWORD '13456'
CONNECTION LIMIT 2
VALID UNTIL '2026-02-27';
```

Explicación:

- `CREATE ROLE`: crea un nuevo rol.
- `LOGIN`: permite iniciar sesión.
- `PASSWORD`: asigna contraseña.
- `CONNECTION LIMIT`: limita conexiones simultáneas.
- `VALID UNTIL`: define la fecha de expiración.

Después de ejecutar el comando, verifiqué que el usuario se creó correctamente en el apartado *Login/Group Roles*.



III. ACTIVIDAD 2: CREAR UN ROL Y ASIGNAR PERMISOS

La segunda actividad consistió en crear un rol con permisos sobre una tabla llamada estudiante. Los permisos solicitados fueron:

- Lectura (SELECT)
- Actualización (UPDATE)
- Borrado (DELETE)

Además, debía asignar ese rol al usuario creado en la actividad anterior.

III-A. Paso 1: Crear la tabla estudiante

Primero creé una tabla de prueba:

```
CREATE TABLE estudiante (
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(50),
    carrera VARCHAR(50)
);
```

III-B. Paso 2: Crear el rol

```
CREATE ROLE rol_estudiante;
```

Este rol no tiene login, solo sirve para agrupar permisos.

III-C. Paso 3: Asignar permisos al rol

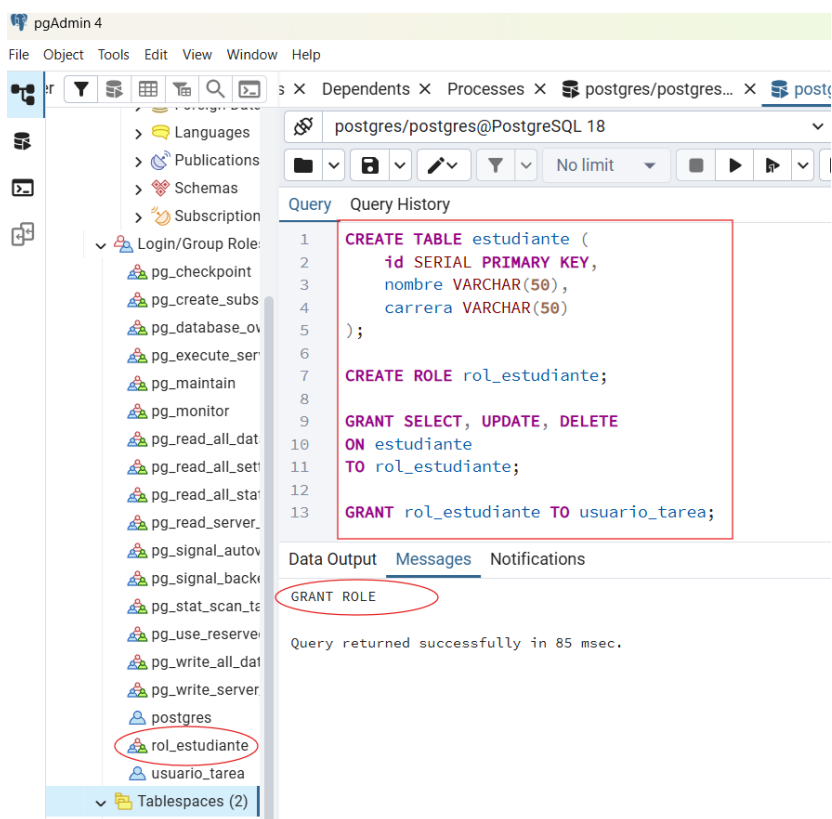
```
GRANT SELECT, UPDATE, DELETE
ON estudiante
TO rol_estudiante;
```

Con esto el rol puede leer, modificar y borrar registros de la tabla.

III-D. Paso 4: Asignar el rol al usuario

```
GRANT rol_estudiante TO usuario_prueba;
```

De esta forma, el usuario creado en la actividad anterior hereda los permisos del rol.



III-E. Importancia de los roles

Los roles permiten organizar los permisos dentro de una base de datos. En lugar de asignar privilegios directamente a cada usuario, se pueden agrupar en roles y luego asignarlos a múltiples usuarios. Esto facilita la administración y mejora la seguridad del sistema.

REFERENCIAS

- [1] PostgreSQL Global Development Group, "Postgresql: Documentation: 18: Grant," PostgreSQL Documentation, 2026, consultado: 17 de febrero de 2026. [Online]. Available: <https://www.postgresql.org/docs/current/sql-grant.html>
- [2] —, "Postgresql: Documentation: 18: Create role," PostgreSQL Documentation, 2026, consultado: 17 de febrero de 2026. [Online]. Available: <https://www.postgresql.org/docs/current/sql-createrole.html>
- [3] IBM, "Gestión de roles de base de datos," IBM Documentation, s.f., consultado: 17 de febrero de 2026. [Online]. Available: <https://www.ibm.com/docs/es/data-studio/4.1.1?topic=management-database-roles>