Yañez Barajas Brandon
Grupo:01
Bases de Datos
Tarea 2

1. ¿Qué requiero para conectarme a una base de datos?

Para conectarme a una base de datos en PostgreSQL, primero necesito asegurarme de que el servidor esté instalado y en ejecución, ya sea en mi computadora local, en un servidor institucional o en la nube. Sin el motor activo, no puedo establecer ninguna conexión.
Después, debo crear un activo o configuración de conexión proporcionando los datos básicos de acceso. Para ello requiero:

- Nombre de la base de datos.
- Host o dirección IP del servidor.
- Puerto (generalmente 5432).
- Mi usuario.
- Mi contraseña.

Si el servidor lo exige, también debo incluir un certificado SSL para una conexión segura. En caso de que la base de datos esté protegida por firewall o no esté expuesta a Internet, necesito configurar conectividad privada o un proxy.
Además, necesito una herramienta cliente desde donde realizar la conexión y ejecutar consultas, como la consola psql o pgAdmin, que me permiten explorar tablas, ejecutar sentencias SQL y administrar los objetos.

2. Permisos a nivel sistema y objeto

En PostgreSQL, el control de acceso se gestiona mediante roles. Un rol puede actuar como usuario o grupo, y es al rol al que se le asignan los privilegios para determinar qué acciones puedo ejecutar y sobre qué recursos.

-Permisos a nivel sistema (globales)

Son privilegios que afectan a todo el servidor o clúster de bases de datos, no a un objeto específico. Me permiten realizar tareas administrativas o generales.

Como usuario, con estos permisos puedo:

- Conectarme al servidor (LOGIN)

- Crear bases de datos (CREATEDB)
- Crear otros roles o usuarios (CREATEROLE)
- Ser superusuario (SUPERUSER)
- Replicar datos (REPLICATION)
- Omitir límites de recursos (BYPASSRLS)

Estos privilegios se asignan al rol completo y aplican de forma global.

-Permisos a nivel objeto (específicos)

Son privilegios que controlan qué operaciones puedo realizar sobre objetos concretos dentro de una base de datos, como tablas, vistas, esquemas o funciones.

Dependiendo del objeto, puedo tener permisos como:

- SELECT (consultar datos)
- INSERT (agregar registros)
- UPDATE (modificar)
- DELETE (eliminar)
- CREATE (crear objetos dentro de un esquema o base)
- EXECUTE (ejecutar funciones o procedimientos)
- REFERENCES (usar claves foráneas)
- USAGE (usar esquemas o secuencias)

3. ¿Cómo dar/quitar permisos?

En postgreSQL puedo asignar o revocar permisos directamente ejecutando sentencias SQL desde la consola `psql` o desde pgAdmin.

PostgreSQL utiliza principalmente dos comandos:

- `GRANT` → para dar permisos

- REVOKE → para quitar permisos

Estos permisos pueden ser a nivel sistema (roles) o a nivel objeto (tablas, vistas, funciones, esquemas, etc.).

4. Diferencia entre role y usuario:

Cuando se habla de un rol, es cuando nos referimos a una entidad que sirve para agrupar y administrar permisos.
Es decir, un rol:

- Puede tener privilegios
- Puede ser dueño de objetos
- Puede asignarse a otros roles
- Puede funcionar como grupo

Normalmente se usa para organizar permisos.

Mientras que el usuario es básicamente un rol que puede conectarse (LOGIN) al sistema. Es decir, representa a una persona o aplicación que accede a la base de datos.

Ejemplo de lo investigado:

-Primero abro pgAdmin 4
-Luego en el panel de Servers le doy doble click e ingresamos la contraseña, te aparecerá algo así:

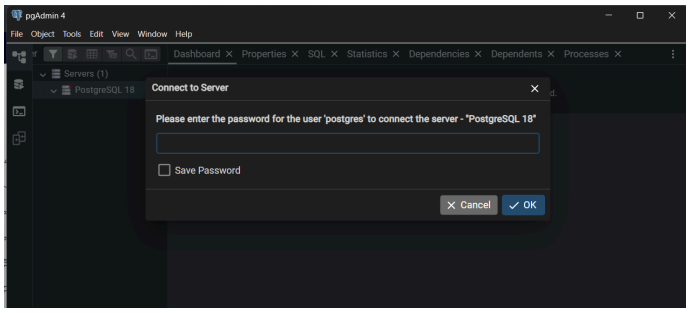

Fig 1. Conexión con el server.

-Ahora ya estoy conectado:

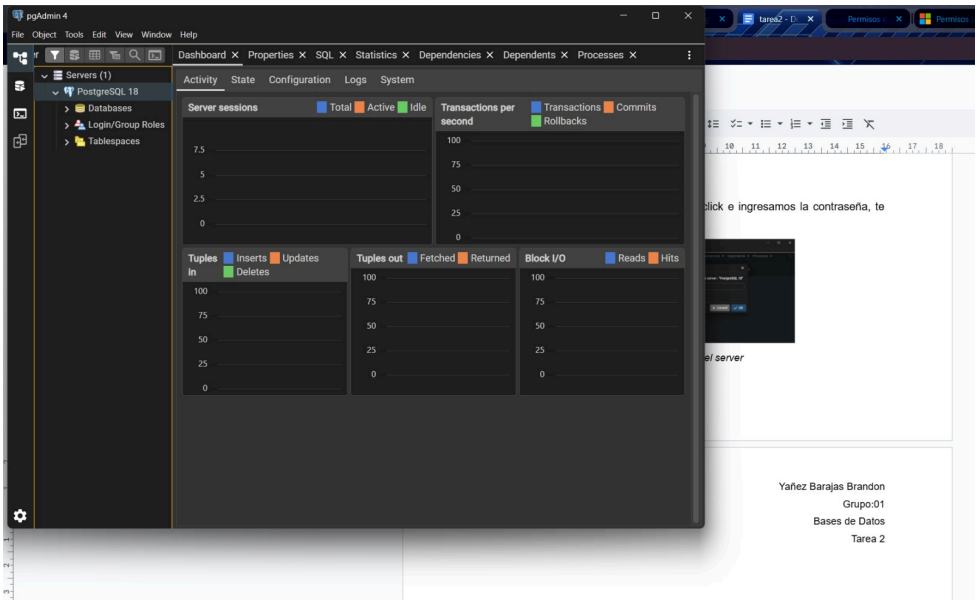

Fig 2. Conexión con la base de datos.

-Posteriormente creo la base de datos: Le damos click derecho en Database → Create → Database. Agregamos el nombre de nuestra base y lo guardamos con save.

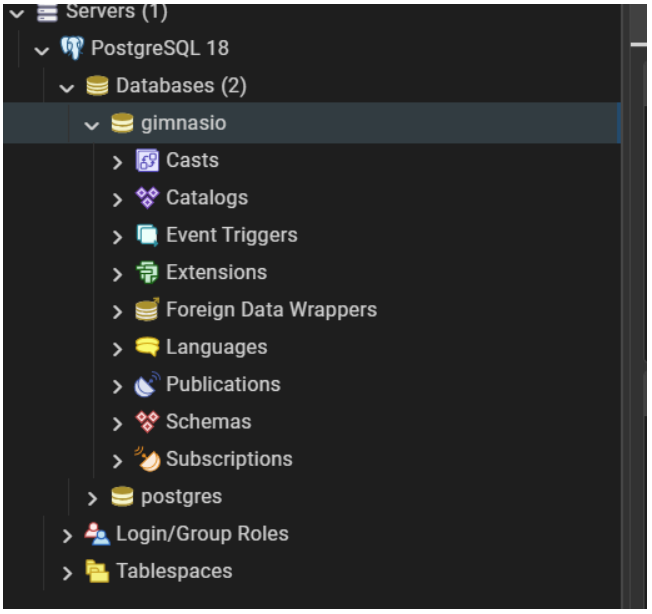

Fig 3. Creación de la base de datos “gimnasio”.

Yañez Barajas Brandon
Grupo:01
Bases de Datos
Tarea 2

-Entro a: Databases → gimnasio → Schemas → public → Tables y le doy click derecho. Te aparecerá lo siguiente:

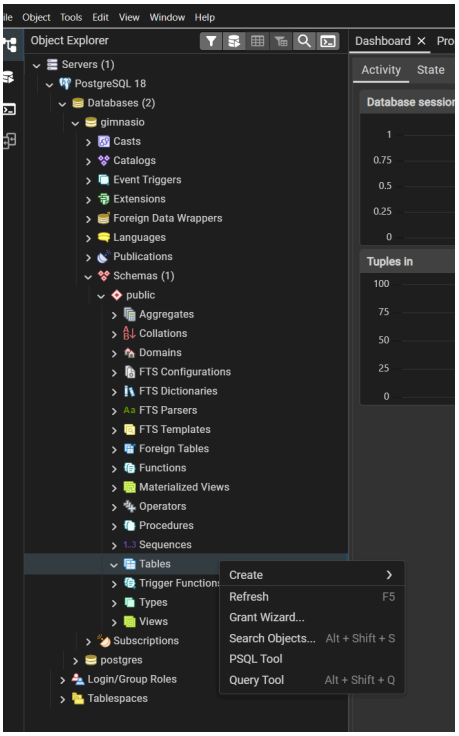

Fig 4. Ubicación para la creación de tablas.

-Creamos la tabla y la guardamos:

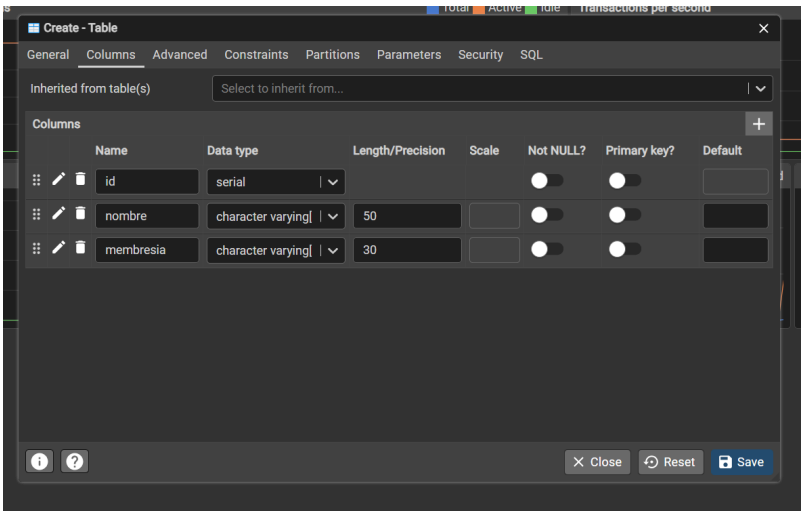

Fig 5.Ejemplo de creación de la tabla.

-Ahora crearemos los roles y usuarios.

-Para la creación de roles nos vamos a Login/Group Role, click derecho y Create → Login/Group Role y creamos dos roles, uno para entrenadores y otro para administradores.

Fig 6. Creación del Rol entrenadores y administradores.

-Para crear los usuarios (roles con LOGIN) realizamos lo mismo solo que activamos el login y le ponemos una contraseña ahorita como simulación.

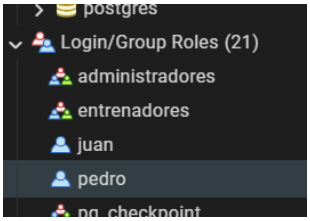

Fig 7. Creación de usuarios.

-Posteriormente, se asignan los **privilegios globales**: se edita el usuario *pedro* dando clic derecho sobre él → Properties..,, se accede a la pestaña *Privileges*, se activan las opciones mostradas en la Figura 8 y, finalmente, se guardan los cambios presionando *Save*.

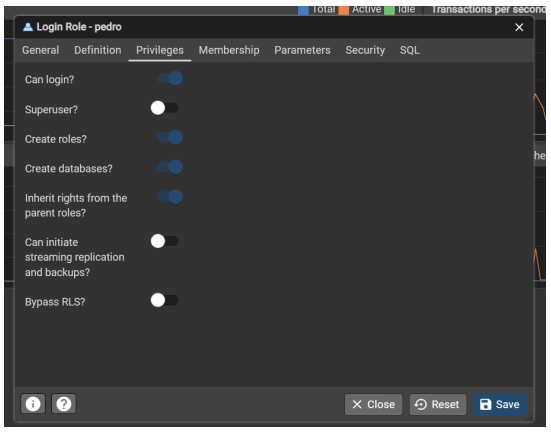

Fig 8. Privileges: Create databases and roles.

-Una vez realizado lo anterior, pasamos a los **permisos a nivel objeto** que es sobre la tabla de clientes. Por ejemplo para entrenadores (solo lectura). Le damos click

derecho en la tabla clientes→ Properties→Security→ Add→ Role: entrenadores→ Marco: SELECT y guardamos así como se muestra en la fig 8.

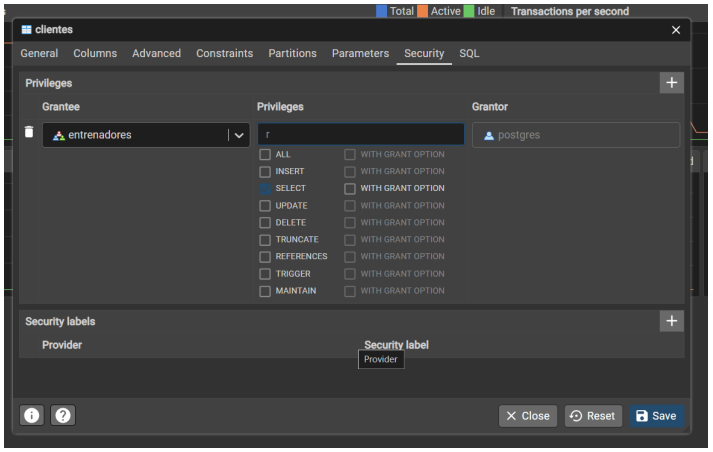

Fig 9. Permisos a nivel objeto de entrenadores.

-Para los administradores realizamos lo mismo que lo anterior, solo que marcamos ALL y guardamos.

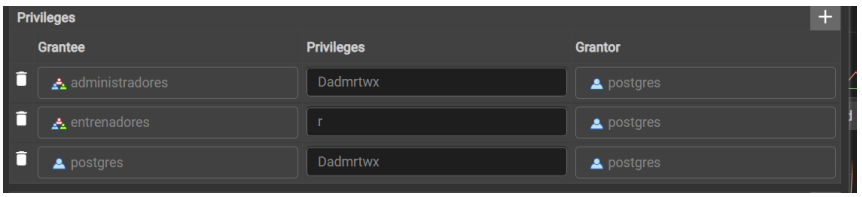

Fig 10. Permisos a nivel objeto de administradores.

-Asignar usuarios a roles

-Abrimos a juan → Membership→ Add row en member of y agrego entrenadores así como se muestra en la fig 10. Luego abro a pedro realizó lo mismo y guardamos.

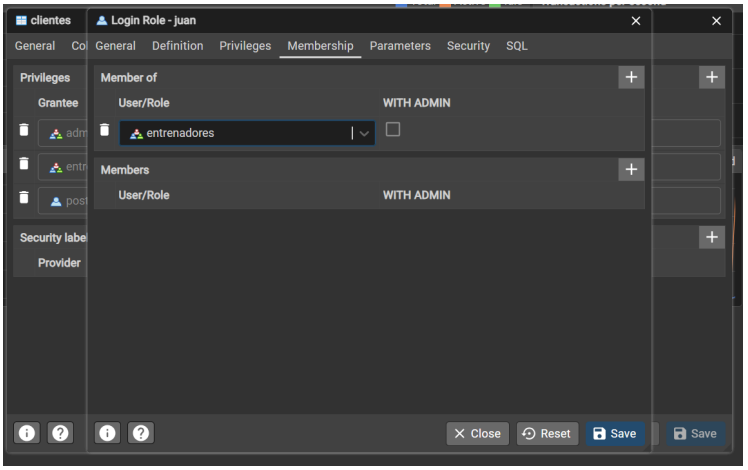

Fig 11. Ejemplo de asignar usuarios a roles.

-Finalmente si quiero quitar permisos, nos vamos a clientes que es mi tabla→ Properties Security → Privileges → selecciono el rol → desmarco permisos o elimino la fila.

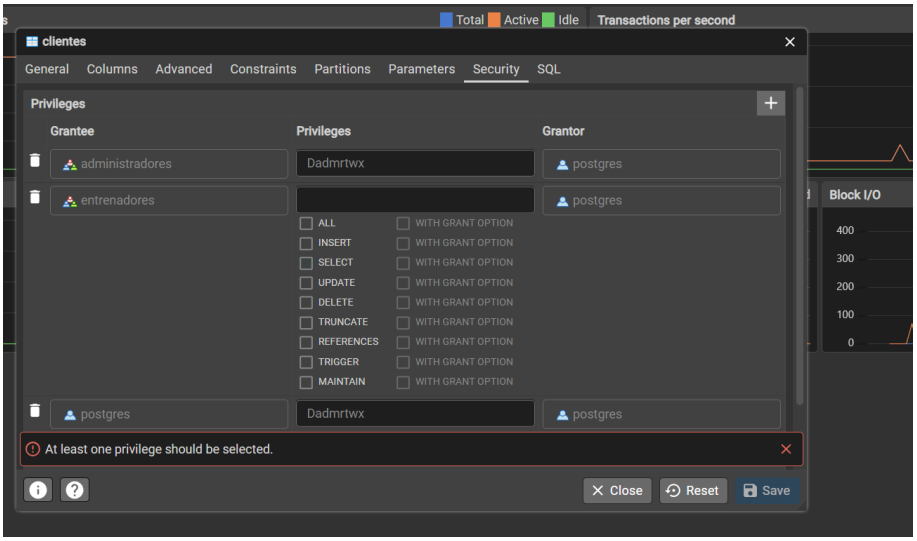

Fig 12. Ejemplo de eliminación de permisos.

Como conclusión, en esta tarea trabajé la administración de seguridad en PostgreSQL utilizando la interfaz gráfica de pgAdmin, donde aprendí a conectarme al servidor, crear la base de datos y definir tanto roles como usuarios. Comprendí que los roles funcionan como grupos para organizar permisos y que los usuarios con LOGIN son quienes realmente acceden al sistema. A partir de ello, asigné privilegios a nivel sistema (como crear bases o roles) y permisos a nivel objeto (SELECT, INSERT, UPDATE, DELETE sobre tablas específicas), además de gestionar membresías para heredar permisos de forma más eficiente. También practiqué la revocación de accesos cuando fue necesario, reforzando el control de seguridad. Finalmente, entendí que toda esta configuración puede hacerse no solo desde la interfaz gráfica, sino también directamente mediante sentencias SQL como CREATE ROLE, GRANT, REVOKE y ALTER ROLE, lo que brinda mayor flexibilidad, automatización y control en entornos profesionales.

Yañez Barajas Brandon
Grupo:01
Bases de Datos
Tarea 2

Bibliografía

- IBM. Conexión de PostgreSQL, IBM Documentation. Disponible: https://www.ibm.com/docs/es/ws-and-kc?topic=connectors-postgresql-connection.
- IBM. Permisos de objeto y sistema, IBM Documentation. Disponible :https://www.ibm.com/docs/es/netcoolomnibus/8.1.0?topic=roles-system-object-permissions.
- IBM. Privilegios de base de datos PostgreSQL, IBM Documentation. Disponible:https://www.ibm.com/docs/es/baw/23.0.x?topic=privileges-postgresql-database.
- Microsoft. Permisos (motor de base de datos) – SQL Server, Microsoft Learn. Disponible:https://learn.microsoft.com/es-es/sql/relational-databases/security/permissions-database-engine?view=sql-server-ver17.
- PostgreSQL Global Development Group. Database Roles and Privileges, PostgreSQL 8.1 Documentation. Disponible: https://www.postgresql.org/docs/8.1/user-manag.html.