Imagen que contiene grafiti, dibujo

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente con confianza mediaInstituto Politécnico Nacional

Escuela Superior de Ingeniería Mecánica y Eléctrica  
Unidad Culhuacán

Ingeniería en computación

Base de Datos

Manual de usuario para la creación de usuarios,

manejo de roles y permisos en PostgreSQL

Alumno: López Sánchez Luis Ricardo

Profesora: Jesús Mora Jain

06 de diciembre 2022 Grupo: 7CV23

**Índice**

[1.- Introducción 5](#_Toc121192468)

[2.- Objetivo del documento 6](#_Toc121192469)

[3.- Funcionalidad 6](#_Toc121192470)

[4.- Creación base de datos de prueba 7](#_Toc121192471)

[5.- Configurar esquema 8](#_Toc121192472)

[6.- Creación de usuario 9](#_Toc121192473)

[7.- Esquema public y rol public 13](#_Toc121192474)

[8.- Administración de permisos en PostgreSQL 14](#_Toc121192475)

[9.- Modificación en los roles 16](#_Toc121192476)

[10.- Asignación de permisos 17](#_Toc121192477)

[11.- Revocación de los permisos 18](#_Toc121192478)

[12.- Inicio de sesión en BD 19](#_Toc121192479)

[13.- Asignación de permisos en un esquema 21](#_Toc121192480)

[14.- Conclusión 22](#_Toc121192481)

[15.- Glosario 23](#_Toc121192482)

[16.- Bibliografía 23](#_Toc121192483)

**Índice de imágenes**

[Imagen 1: Símbolo SQL Shell 7](file:///C:\Users\ricar\Documents\7MO%20SEMESTRE\BaseDeDatos\Parcial%20tres\Final\Luis_Ricardo_LopezSanchez_Final.docx#_Toc121192275)

[Imagen 2: Dentro de la consola SQL Shell 7](#_Toc121192276)

[Imagen 3: Creación BD pruebas 8](#_Toc121192277)

[Imagen 4: Creación esquema 8](#_Toc121192278)

[Imagen 5: Listado de usuarios 9](#_Toc121192279)

[Imagen 6: Creación de usuario en SQL Shell 10](#_Toc121192280)

[Imagen 7: Creación de usuario en pgAdmin 10](#_Toc121192281)

[Imagen 8: Ventana para crear rol pgAdmin (Sección general) 11](#_Toc121192282)

[Imagen 9: Ventana para crear rol pgAdmin (Sección Definition) 11](#_Toc121192283)

[Imagen 10: Ventana para crear rol pgAdmin (Sección Privileges) 12](#_Toc121192284)

[Imagen 11: Ventana para crear rol pgAdmin (sección SQL) 12](#_Toc121192285)

[Imagen 12: Rol creado mediante pgAdmin 13](#_Toc121192286)

[Imagen 13: Eliminando permisos esquema público 14](#_Toc121192287)

[Imagen 14: Creación usrtest2 15](#_Toc121192288)

[Imagen 15: Rol readonly sin permisos 16](#_Toc121192289)

[Imagen 16: Error de Login 17](#_Toc121192290)

[Imagen 17: Solución error Login 17](#_Toc121192291)

[Imagen 18: Asignación de permiso usrtest 17](#_Toc121192292)

[Imagen 19: Asignación de permisos 18](file:///C:\Users\ricar\Documents\7MO%20SEMESTRE\BaseDeDatos\Parcial%20tres\Final\Luis_Ricardo_LopezSanchez_Final.docx#_Toc121192293)

[Imagen 20: Revocación de permisos 18](#_Toc121192294)

[Imagen 21: Cambio de conexión entre bases de datos 19](#_Toc121192295)

[Imagen 22: Error de conexión en BD 19](#_Toc121192296)

[Imagen 23: Conexión a BD pruebas con el usuario (usrTest2) 20](#_Toc121192297)

[Imagen 24: Listado BD en pgAdmin 20](#_Toc121192298)

[Imagen 25: Permisos de uso en un esquema 21](#_Toc121192299)

# 1.- Introducción

PostgreSQL es uno de los sistemas de bases de datos relacionales de código abierto más populares. Con más de 30 años de trabajo de desarrollo, PostgreSQL ha demostrado ser una base de datos altamente fiable y robusta que puede manejar un gran número de cargas de trabajo de datos complejas.

Icon

Description automatically generated

Algo importante de aclarar es que dentro de este gestor de Base de Datos (BD) los usuarios, grupos y roles son lo mismo en PostgreSQL, y la única diferencia es que los usuarios tienen permiso para iniciar sesión de forma predeterminada. Las instrucciones *CREATE USER* y *CREATE GROUP* son en realidad alias de la instrucción *CREATE ROLE*..

# 2.- Creación de usuario

Antes de pasar a la creación del usuario, se hará uso del comando *\du+* para que se listen los usuarios que tiene creados dentro de su dispositivo (Ver Imagen 5).

Text

Description automatically generated

Imagen : Listado de usuarios

En este ejemplo se muestran algunos usuarios más aparte del super usuario *postgres* en su caso el único usuario que debe aparecer es *postgres*. En esta tabla se muestran el nombre del usuario, los *roles* que tiene asignados entre otros atributos (Ver Imagen 5).

Ahora para poder crear un usuario es necesario escribir la siguiente instrucción (Ver Imagen 6):

CREATE USER nombre\_usuario WITH PASSWORD ‘contraseña’;

También se puede utilizar la siguiente instrucción, pues como se dijo en la introducción un *usuario* es igual a un *Role*:

CREATE ROLE nombre\_usuario WITH LOGIN PASSWORD ‘contraseña’

Como se ve en la Imagen 6 se escribe esta instrucción y aparece el *usuario* recientemente creado más los que ya se tenían previamente creados. Es importante notar que el usuario recién creado no tiene ningún tipo de permiso.

Text

Description automatically generated

Imagen : Creación de usuario en SQL Shell

¡Listo!, con esta breve explicación se pueden crear *usuarios* desde la consola en secciones posteriores se mostrará como es que se asignan roles a nuestros usuarios.

Ahora crear un *usuario* en pgAdmin 4 es más sencillo, basta con ir a la sección *Login/Group Roles* para crear un nuevo rol. Localizada esta sección debemos dar click derecho sobre esta, luego dar en créate y por último dar click en *Login/Group Role* (Ver imagen 7).

Graphical user interface, text, application

Description automatically generated

Imagen : Creación de usuario en pgAdmin

Seguidos los pasos anteriores se desplegara un ventana como la de la Imagen 8. En esta ventana aparecen 7 apartados distintos, estos son:

* General: es donde colocaremos el nombre de nuestro nuevo rol

Graphical user interface, application, Word

Description automatically generated

Imagen : Ventana para crear rol pgAdmin (Sección general)

* Definition: En este apartado definimos la contraseña del usuario, si tiene un límite de tiempo para ser usado y el límite de conexiones que tiene

Graphical user interface, text, application, Teams

Description automatically generated

Imagen : Ventana para crear rol pgAdmin (Sección Definition)

* Privileges: En esta ventana se asignan los permisos que va a tener este nuevo rol

Chart

Description automatically generated

Imagen : Ventana para crear rol pgAdmin (Sección Privileges)

* Las secciones de Membership, Parameters y Security son secciones que para efectos de este manual no son importantes
* SQL: en esta sección nos muestra el código que sirve para generar este nuevo usuario

Graphical user interface, application

Description automatically generated with medium confidence

Imagen : Ventana para crear rol pgAdmin (sección SQL)

Al dar click al botón de Save este nuevo usuario se guardara dentro de PostgreSQL tal como lo muestra la siguiente imagen.

Table

Description automatically generated

Imagen : Rol creado mediante pgAdmin

De estas dos maneras es como se crean usuarios dentro de PostgreSQL usando la consola y el programa PgAdmin 4.

# 3.- Esquema public y rol public

Cuando se crea una nueva base de datos, PostgreSQL crea de forma predeterminada un esquema denominado *public*. A todos los *usuarios* y *roles* nuevos se les concede de forma predeterminada el rol *public* y, por lo tanto, pueden crear objetos en el esquema *public*.

Esto se convierte en un problema si intenta crear un usuario de solo lectura. Incluso si restringe todos los privilegios, los permisos heredados a través del *rol* *public* permiten al usuario crear objetos en el esquema *public*.

Para solucionarlo, se debe revocar el permiso de creación predeterminado en el esquema *public* desde el rol *public* mediante la siguiente instrucción SQL:

REVOKE CREATE ON SCHEMA public FROM PUBLIC;

La siguiente declaración revoca la capacidad del rol público de conectarse a la base de datos:

REVOKE ALL ON DATABASE nom\_BD FROM PUBLIC;

Esto garantiza que los usuarios no puedan conectarse a la base de datos de forma predeterminada a menos que se conceda explícitamente este permiso.

La revocación de los permisos del *rol* *public* afecta a todos los *usuarios* y *roles* existentes.

Los *usuarios* y *roles* que deberían de poder conectarse a la base de datos o crear objetos en el esquema público deben recibir los permisos explícitamente antes de revocar los permisos del *rol* *public* en el entorno de producción.

Escriba dentro de SQL Shell los comandos anteriores (Ver Imagen 13).

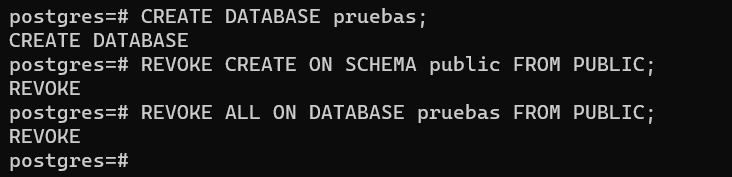


Imagen : Eliminando permisos esquema público

Con lo anterior garantizamos que los usuarios no puedan conectarse a la base de datos de forma predeterminada a menos que se conceda explícitamente este permiso.

La revocación de los permisos del *rol* *public* afecta a todos los usuarios y roles existentes.

Una vez hecho esto se mostrará como es la creación de *roles* dentro de SQL Shell

# 4.- Administración de permisos en PostgreSQL

Dentro de PostgreSQL existen distintos permisos para la gestión de la BD, algunos de estos permisos se muestran en la Tabla 4.

Tabla : Permisos en PostgreSQL

|  |  |
| --- | --- |
| Permisos en PostgreSQL | |
| PERMISO | DESCRIPCIÓN |
| LOGIN/ NOLOGIN | Permitir (o no) iniciar sesión en PostgreSQL |
| SUPERUSER/ NOSUPERUSER | Permite (o no) permisos de superusuario (\*) |
| CREATEDB/ NOCREATEDB | Permite (o no) la capacidad de crear nuevas bases de datos |
| CREATEROLE/ NOCREATEROLE | Permite (o no) la capacidad de crear nuevos roles |
| CREATEUSER/ NOCREATEUSER | Permite (o no) la capacidad de crear nuevos usuarios |
| INHERIT/ NOINHERIT | Permite (o no) la capacidad de hacer que los privilegios se pueda heredar. |
| REPLICATION/ NOREPLICATION | Concede (o no) permisos de replica |

\*Un superusuario de la base de datos omitirá otras verificaciones de permisos, excepto LOGIN (debe otorgarse por separado)

En la Imagen 8 se muestra la creación de un nuevo usuario de prueba al cual le vamos a asignar permisos en secciones posteriores.

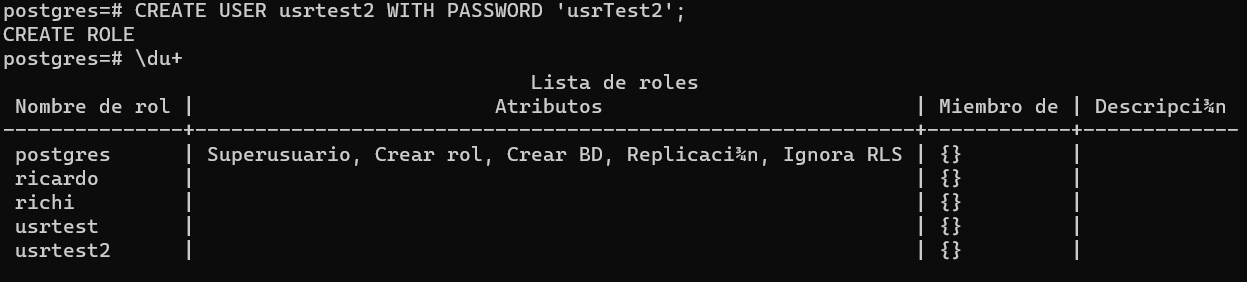


Imagen : Creación usrtest2

Para conceder permisos a los usuarios, lo primero que debe hacer es crear un nuevo *rol*. A este nuevo *rol* le vamos a asignar los permisos que nosotros queramos. Escriba dentro de SQL Shell el siguiente comando:

CREATE ROLE nom\_Role;

Cuando se ejecuta el comando anterior se crea un *role* simple que no tiene ningún tipo de permiso. Para efectos de este manual se creará un *rol* llamado *readonly.*

Al despeglar el comando \du+ observamos como en la columna de *atributos* el rol recién creado (***readonly***) no tiene permitido iniciar sesión, esto se logró al hacer la configuración de la sección anterior.

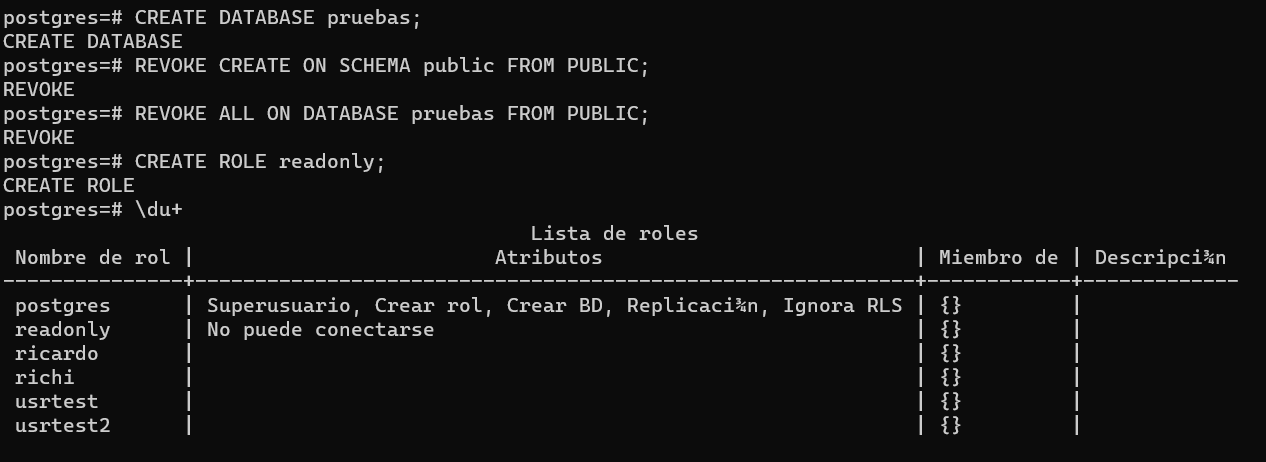


Imagen : Rol readonly sin permisos

Para asignar a este *rol* el permiso de inicio de sesión escriba y ejecute el siguiente comando:

GRANT CONNECT ON DATABASE nom\_BD TO nom\_Role;

Si desea simplificar este paso puede escribir el siguiente comando:

CREATE ROLE nom\_Role WITH permiso1, permiso2;

El comando anterior crea un *rol* y al mismo tiempo le asigna los permisos que se desea que tengan este *rol*.

De esta manera es como podemos crear roles dentro de PostgreSQL.

# 5.- Modificación en los roles

Si utilizamos nuevamente el comando *\du+* y nos aparece como lo resaltado en la Imagen seis, debemos hacer uso del siguiente comando para la modificación de los permisos del role creado en la sección anterior:

ALTER ROLE nom\_Role WITH permiso;

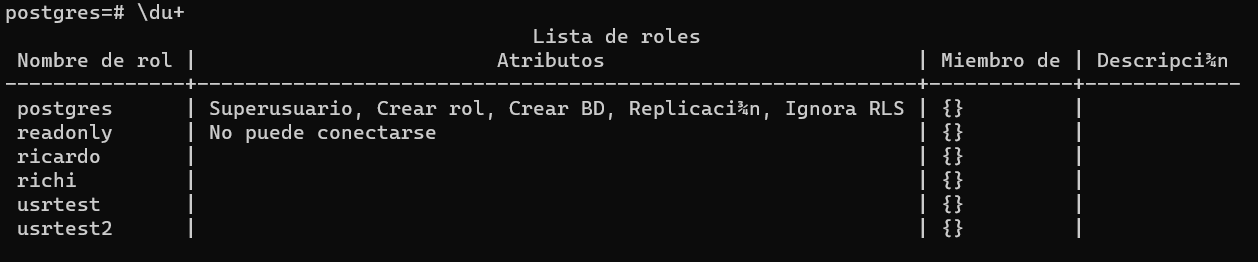


Imagen : Error de Login

Después de aplicar este comando la tabla de roles deberá verse como la Imagen 11:

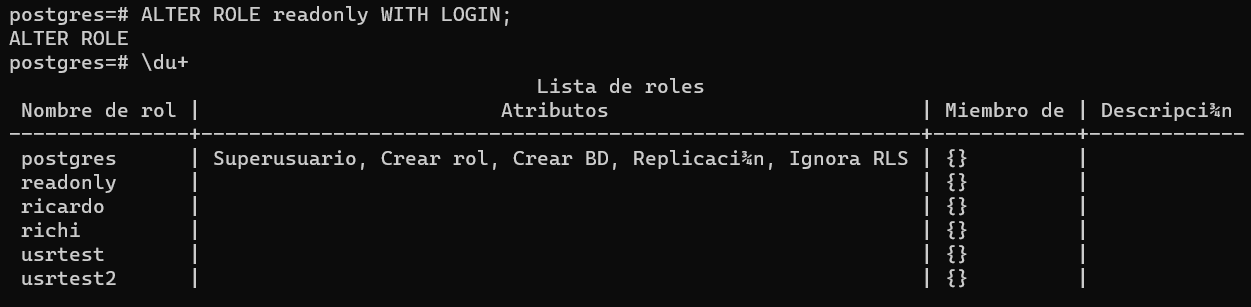


Imagen : Solución error Login

Con este comando es como podemos modificar los permisos que le damos a cada uno de los roles dentro de nuestro gestor de BD.

# 6.- Asignación de permisos

Como se menciona anteriormente dentro de PostgreSQL un *rol* es lo mismo que un *usuario* por lo tanto podemos hacer uso de dos comandos para gestionar los permisos. El primero de es el siguiente:

ALTER ROLE nom\_rol WITH permiso;

Este comando modifica el *rol* que le indiquemos y le concede el permiso que nosotros queramos que tenga este *rol*. (Ver Imagen 12)



Imagen : Asignación de permiso usrtest

Como se observa en la imagen anterior estamos modificando el rol *usrtest* y le estamos dando el permiso para que pueda crear roles

Ahora podemos asignar los permisos del rol *readonly* al usuario *usrtest2* para esto usamos el siguiente comando

GRANT nom\_Rol TO usr\_Nom;

El resultado de aplicar estos dos comandos se muestra en la Imagen 13.

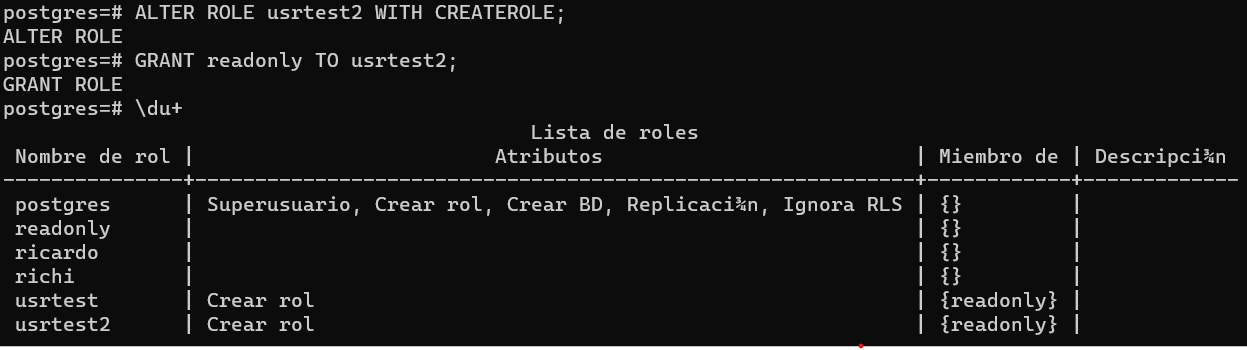


Imagen : Asignación de permisos

# 11.- Revocación de los permisos

Para poder remover o quitar permisos dentro de PostgreSQL es muy sencillo pues basta con escribir el siguiente comando:

REVOKE nom\_Rol FROM nom\_Usr

En la imagen 14 se muestra el uso de este comando y se nota como el *usrtest* ya no em miembro del rol *readonly*.

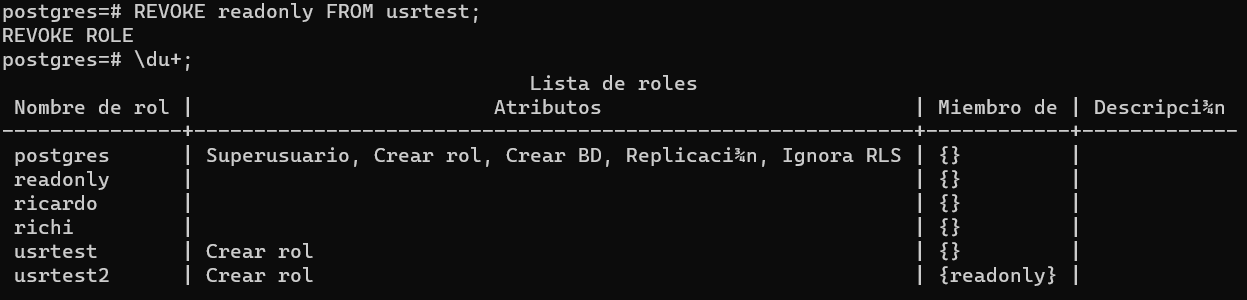


Imagen : Revocación de permisos

# 7.- Inicio de sesión en BD

Cuando se accede a la interfaz psql, se accede directamente a una base de datos. Por lo tanto, cuando accedes ya te encuentras con una base de datos seleccionada. Esto no impide que puedas listar todas las bases de datos del sistema y seleccionar otra.

Para conectarse desde PSQ Shell debemos escribir el siguiente comando:

\connect nom\_BD o \c nom\_Bd

En la siguiente imagen se ve la implementación de este comando dentro de SQL Shell

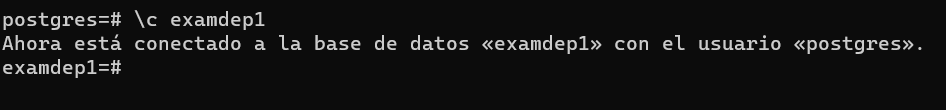


Imagen : Cambio de conexión entre bases de datos

Si desea conectarse a una BD con otro usuario deberá escribir el comando siguiente:

\c nom\_BD nom\_Usr

Es importante verificar que el usuario con el que deseamos conectarnos a es BD tengas el permiso de inicio de sesión, de lo contrario esto provocara un error como el que se ve en la Imagen 16

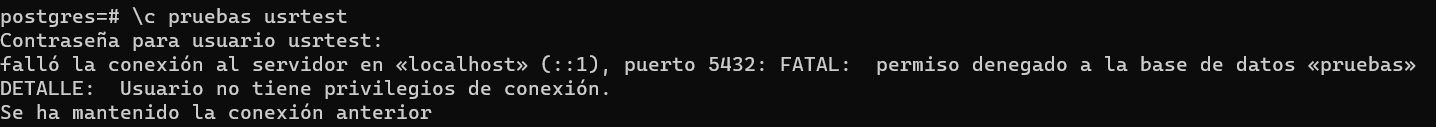


Imagen : Error de conexión en BD

Si desea asignar permisos de login en una BD en particular ve a la sección número nueve de este manual.

En la sección nueve de este manual se creo un usuario (usrTest2) al cual se le concedieron los permisos de login en la BD *pruebas* que si tratamos de conectarnos con este usuario no nos dará ningún conflicto (Ver Imagen 17).

Text

Description automatically generated

Imagen : Conexión a BD pruebas con el usuario (usrTest2)

De esta manera es como podemos conectarnos a una BD mediante el uso de SQL Shell.

Ahora usando pgAdmin 4 es más sencillo.

En la parte izquierda de pgAdmin aparecen listadas todas las BD que se encuentran dentro de nuestro navegador. Para cambiar de conexión entre BD basta con dar doble click en donde indica la flecha verde y de esta manera nos conectaríamos a una BD distinta a la de *postgre* (Ver imagen 18).

Graphical user interface, application

Description automatically generated with medium confidence

Imagen : Listado BD en pgAdmin

Como se observa en la imagen anterior todas las base de datos con las que no nos encontremos conectados aparecerán con un pequeña cruz de color rojo como lo indica la flecha azul. De esta manera sabemos que BD es con las que tenemos una conexión.

# 8.- Asignación de permisos en un esquema

Para asignar permisos de uso de un rol en un esquema determinado es necesario seguir y escribir las líneas de código que se muestran en la Imagen 25

Graphical user interface, text, application, email

Description automatically generated

Imagen : Permisos de uso en un esquema

En la primer línea de código lo que se hace es crear un nuevo rol llamado *readonly2.*

En la línea numero cuatro le asignamos el permiso de conexión a este rol dentro de la base de datos (*pruebas*) creada en secciones anteriores.

En la línea número siete le otorgamos el permiso para que este rol se pueda usar en esquema *pruebas\_sc* el cual fue igualmente creado en secciones anteriores

Por último, al ser un rol solo de lectura, le otorgamos que pueda hacer uso de la instrucción *SELECT* a todas las tablas que contenga este esquema.

Siguiendo todos estos pasos se logra la asignación de permisos a un rol dentro de un esquema especifico.

# 9.- Conclusión

A lo largo de este documento se explico como es que se recomienda la gestión de usuarios y roles dentro de PostgreSQL en su versión 14.

Se puede concluir que al término de la lectura de este manual el usuario va a ser capaz de administrar de una eficiente y segura el acceso a la información que se maneja en su base de datos. Evitando accesos no deseados, modificaciones no deseadas, inicios de sesión no deseados entre otras factores que se mencionan a lo largo de este documento.

Además de la creación de usuarios, roles, bases de datos y esquemas dentro estas para que la manera en la que se organiza su información sea de una mejor manera y al mismo tiempo sea segura.

# 10.- Bibliografía

Perdiz, Á. (2020b, junio 1). Gestionar permisos en PostgreSQL. Álvaro Perdiz. <https://alvaroperdiz.com/bases-de-datos/postgresql/como-gestionar-permisos/>

Perdiz, Á. (2020a, junio 1). Cambiar entre bases de datos en PostgreSQL. Álvaro Perdiz. <https://alvaroperdiz.com/bases-de-datos/postgresql/como-cambiar-entre-bases-de-datos/>

Administración de usuarios y roles de PostgreSQL. (2021, 19 agosto). Amazon Web Services. <https://aws.amazon.com/es/blogs/aws-spanish/managing-postgresql-users-and-roles/>

-, D. (2019, 24 octubre). Cómo crear un usuario y asignarle permisos en PostgreSQL. DatabaseAndTechnology. Recuperado 6 de diciembre de 2022, de <https://databaseandtech.wordpress.com/2019/10/24/como-crear-un-usuario-y-asignarle-permisos-en-postgresql/>

colaboradores de Wikipedia. (2022, 5 julio). PostgreSQL. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/PostgreSQL>