



Instituto 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

HOJA DE CONTROL

Tabla 1. Información general

| | | | |
|------------------------|---|----------------------------|------------|
| Organismo | Instituto Politécnico Nacional – Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán | | |
| Proyecto | Gestión de usuarios y roles en PostgreSQL | | |
| Entregable | Manuela de Usuario | | |
| Autor | López Sánchez Luis Ricardo | | |
| Versión/Edición | 1.0.0 | Fecha versión | 06/12/2022 |
| | | N° Total de Páginas | 24 |

Tabla 2: Registro de cambios

| Versión | Causa de cambio | Responsable del cambio | Fecha del cambio |
|----------------|------------------------|-------------------------------|-------------------------|
| 1.0.0 | Versión inicial | Luis Ricardo López Sánchez | 06/12/2022 |
| | | | |
| | | | |

Tabla 3: Control distribución

| Nombre y apellidos |
|----------------------------|
| Luis Ricardo López Sánchez |
| |
| |
| |
| |
| |
| |
| |

Índice

| | |
|--|----|
| 1.- Introducción | 5 |
| 2.- Objetivo del documento | 6 |
| 3.- Funcionalidad..... | 6 |
| 4.- Creación base de datos de prueba | 7 |
| 5.- Configurar esquema | 8 |
| 6.- Creación de usuario..... | 9 |
| 7.- Esquema public y rol public..... | 13 |
| 8.- Administración de permisos en PostgreSQL | 14 |
| 9.- Modificación en los roles | 16 |
| 10.- Asignación de permisos | 17 |
| 11.- Revocación de los permisos | 18 |
| 12.- Inicio de sesión en BD | 19 |
| 13.- Asignación de permisos en un esquema..... | 21 |
| 14.- Conclusión..... | 22 |
| 15.- Glosario | 23 |
| 16.- Bibliografía..... | 23 |

Índice de imágenes

| | |
|---|----|
| Imagen 1: Símbolo SQL Shell..... | 7 |
| Imagen 2: Dentro de la consola SQL Shell | 7 |
| Imagen 3: Creación BD pruebas..... | 8 |
| Imagen 4: Creación esquema..... | 8 |
| Imagen 5: Listado de usuarios..... | 9 |
| Imagen 6: Creación de usuario en SQL Shell..... | 10 |
| Imagen 7: Creación de usuario en pgAdmin | 10 |

| | |
|--|----|
| Imagen 8: Ventana para crear rol pgAdmin (Sección general) | 11 |
| Imagen 9: Ventana para crear rol pgAdmin (Sección Definition)..... | 11 |
| Imagen 10: Ventana para crear rol pgAdmin (Sección Privileges) | 12 |
| Imagen 11: Ventana para crear rol pgAdmin (sección SQL) | 12 |
| Imagen 12: Rol creado mediante pgAdmin..... | 13 |
| Imagen 13: Eliminando permisos esquema público..... | 14 |
| Imagen 14: Creación usrtest2 | 15 |
| Imagen 15: Rol readonly sin permisos | 16 |
| Imagen 16: Error de Login | 17 |
| Imagen 17: Solución error Login..... | 17 |
| Imagen 18: Asignación de permiso usrtest..... | 17 |
| Imagen 19: Asignación de permisos..... | 18 |
| Imagen 20: Revocación de permisos | 18 |
| Imagen 21: Cambio de conexión entre bases de datos | 19 |
| Imagen 22: Error de conexión en BD..... | 19 |
| Imagen 23: Conexión a BD pruebas con el usuario (usrTest2)..... | 20 |
| Imagen 24: Listado BD en pgAdmin..... | 20 |
| Imagen 25: Permisos de uso en un esquema | 21 |

Índice de tablas

| | |
|---------------------------------------|----|
| Tabla 1. Información general..... | 2 |
| Tabla 2: Registro de cambios | 2 |
| Tabla 3: Control distribución..... | 2 |
| Tabla 4: Permisos en PostgreSQL | 14 |
| Tabla 5: Glosario | 23 |
| Tabla 6: Bibliografía..... | 23 |

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.



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.- Objetivo del documento

Comprender como es la gestión de los roles y permisos de los usuarios dentro de PostgreSQL utilizando *pgAdmin* o la *consola SQL Shell*.

3.- Funcionalidad

Al término de la lectura de este manual el lector deberá ser capaz de gestionar los usuarios dentro de su base de datos, así como la gestión de permisos que este le asigne en los distintos esquemas que se encuentran dentro de la base de datos para evitar un mal uso en la información contenida en la base datos que sea de su interés.

En este manual de usuario es indiferente el uso de la consola (PSQ Shell) o el programa (pgAdmin 4) para la creación de los roles y asignaciones de permiso pues se demuestra que se pueden usar cualquiera de las dos maneras para llegar la mismo resultado.

4.- Creación base de datos de prueba

En este manual de usuario no se explicará tan a profundidad como es que se crea una Base de Datos pues el objetivo principal de este manual es la gestión de los usuarios dentro de una base de datos.

Para crear una base de datos lo primero que debe realizar para poder crear un usuario (ROLE) es abrir la consola de PostgreSQL que es SQL Shell (psql).

Para hacer esto abra el buscador de Windows y escriba el nombre de esta consola. Si realizó correctamente la instalación de PostgreSQL deberá aparecer un símbolo como el que se muestra en la Imagen 1.

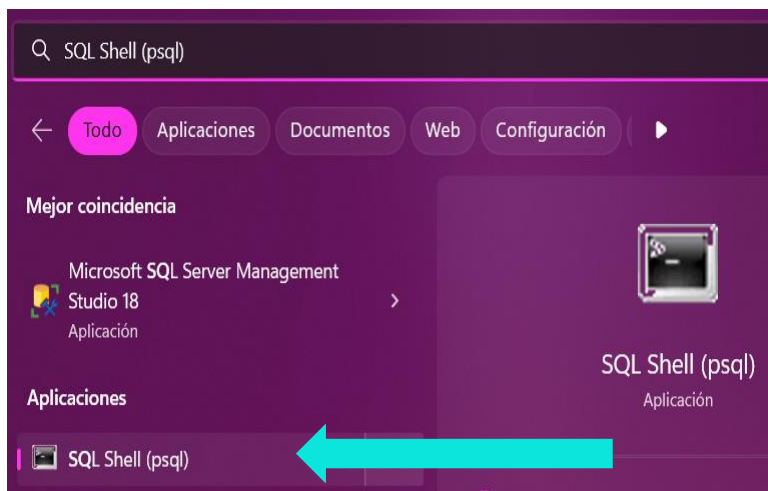


Imagen 1: Símbolo SQL Shell

Ahora, de doble click en donde indica la flecha azul (ver Imagen 1) para ejecutar la consola.

Dentro de la consola es necesario que de Enter hasta que le aparezca la línea (señalada con la flecha azul) que indica que ingrese la contraseña tal y como se muestra en la Imagen dos.

**Nota*:* Esta contraseña fue previamente configurada durante la instalación de PostgreSQL.

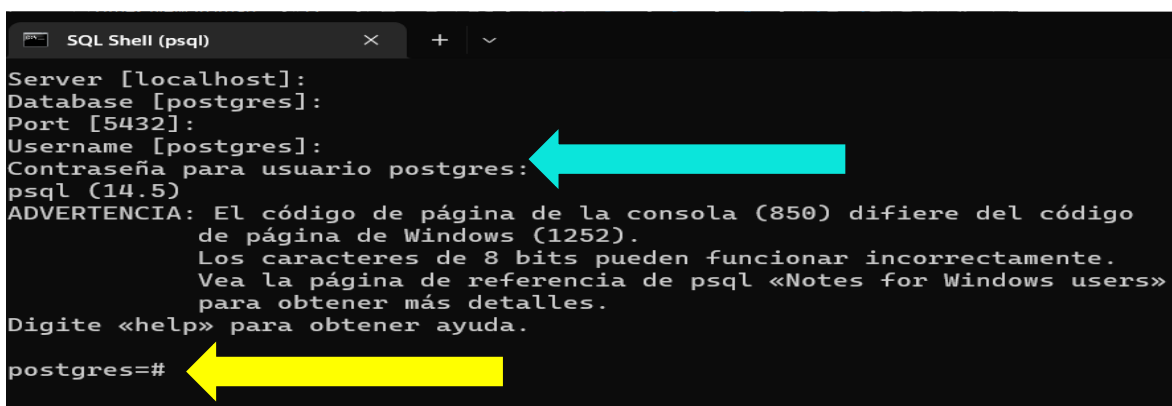


Imagen 2: Dentro de la consola SQL Shell

Si ingreso correctamente su contraseña aparecerá el mensaje anterior y la línea señalada con la flecha amarilla (Ver Imagen 2).

Una vez dentro de la consola escriba el siguiente comando:

CREATE DATABASE nom_BD;

```
postgres=# CREATE DATABASE pruebas;  
CREATE DATABASE  
postgres=#
```

Imagen 3: Creación BD pruebas

Para efectos de este manual se creó una base de datos (BD) llamada *pruebas* como se muestra en la imagen anterior.

5.- Configurar esquema

Una vez creada la BD es necesario configurar un esquema dentro de esta BD de prueba (Ver Imagen 4).

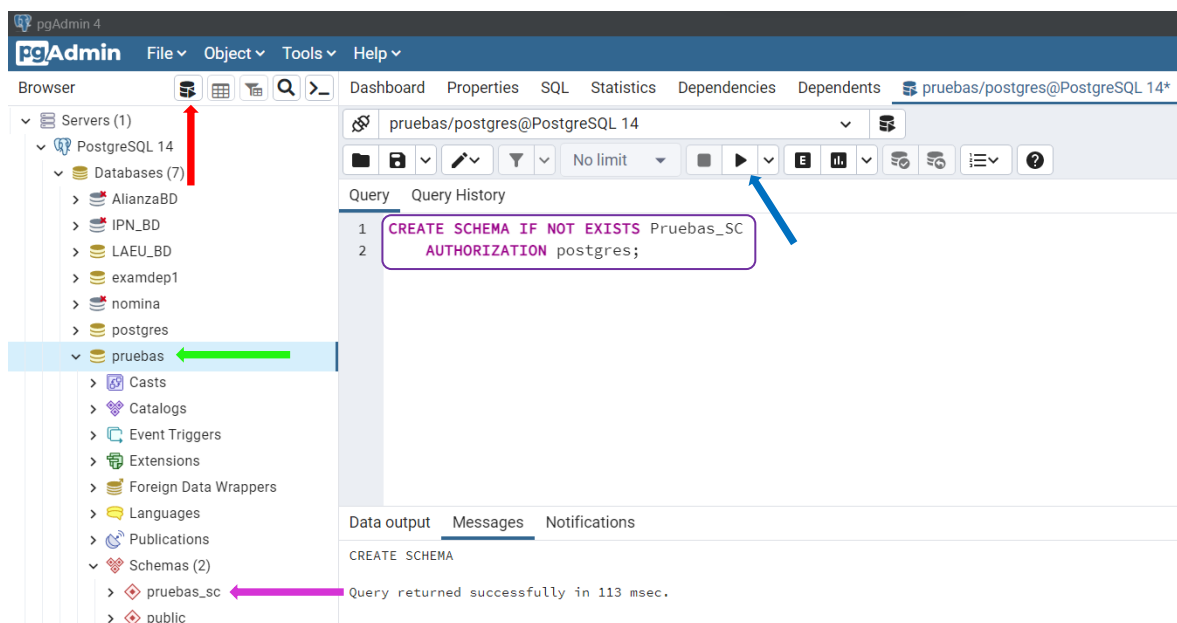


Imagen 4: Creación esquema

Para crear el esquema en la BD pruebas siga los siguientes pasos:

1. Ejecute pgAdmin 4. Una vez dentro de pgAdmin puede verificar que la BD creada en SQL Shell se guardó correctamente (*flecha verde*).

2. Verifique que la BD pruebas se encuentre resalta por un color azul claro. De ser así, de click en onde señala la flecha roja.
3. Escriba el comando(s) encerrados dentro del rectángulo morado.
4. De click para ejecutar el comando en donde señala la flecha azul
5. Si realizo todo correctamente deberá aparecer el esquema creado (flecha rosa).

**Nota*:* Sino aparece el esquema de click derecho sobre el nombre de la BD y de click en la opción *Refresh*. Con esto debería de aparecer el esquema recién creado

Listo, con la BD creada y el esquema creado puede pasar a la creación de usuarios y la gestión de estos dentro de PostgreSQL.

6.- 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).

```
postgres=# \du+
```

| Lista de roles | | | |
|----------------|--|------------|-------------|
| Nombre de rol | Atributos | Miembro de | Descripción |
| dichi | Crear rol | {} | |
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {} | |
| ricardo | | {} | |
| richi | | {} | |

```
postgres=# |
```

Imagen 5: 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.

```
postgres=# CREATE USER usrTest WITH PASSWORD 'usrTest1';
CREATE ROLE
postgres=# \du+
```

| Lista de roles | | | |
|----------------|--|------------|-------------|
| Nombre de rol | Atributos | Miembro de | Descripción |
| dichi | Crear rol | { } | |
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | { } | |
| ricardo | | { } | |
| richi | | { } | |
| <u>usrtest</u> | | { } | |

Imagen 6: 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 *create* y por último dar click en *Login/Group Role* (Ver imagen 7).

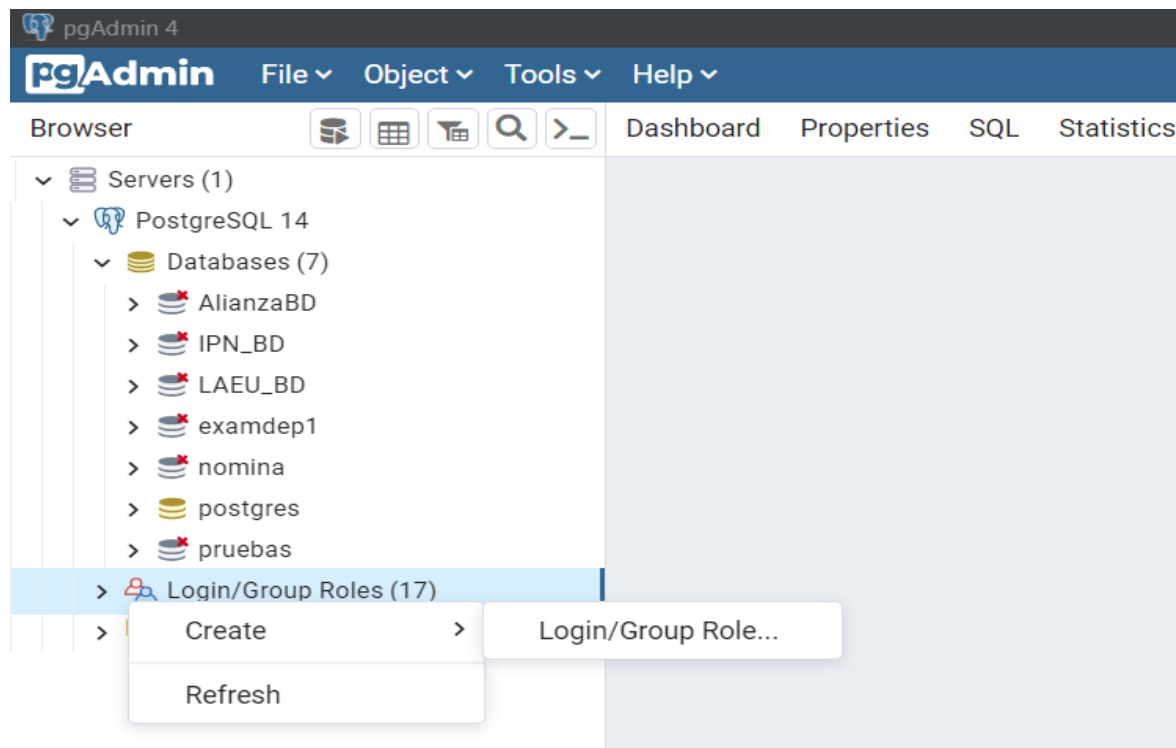


Imagen 7: Creación de usuario en pgAdmin

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

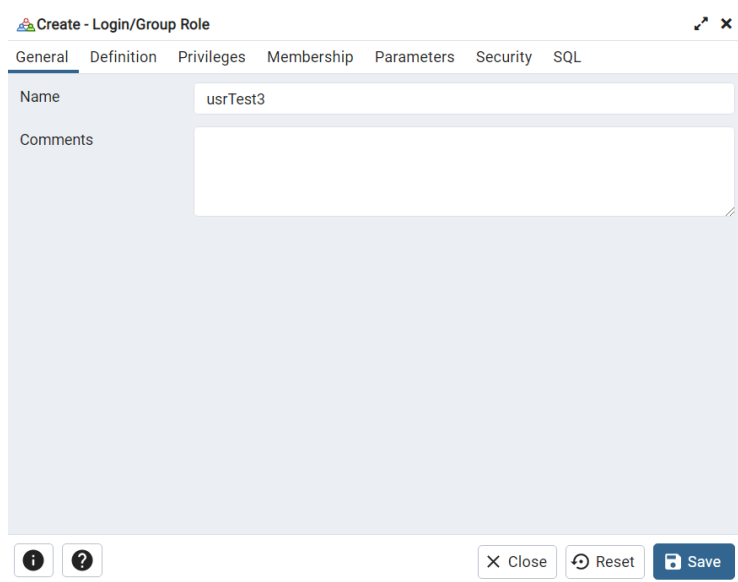


Imagen 8: Ventana para crear rol pgAdmin (Sección general)

- 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

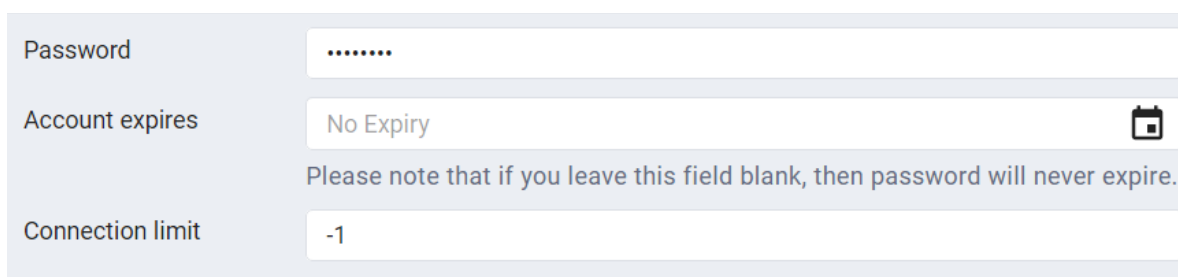


Imagen 9: Ventana para crear rol pgAdmin (Sección Definition)

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

| General | Definition | Privileges | Membership | Parameters | Security | SQL |
|---------|------------|---|-------------------------------------|------------|----------|-----|
| | | Can login? | <input checked="" type="checkbox"/> | | | |
| | | Superuser? | <input type="checkbox"/> | | | |
| | | Create roles? | <input checked="" type="checkbox"/> | | | |
| | | Create databases? | <input type="checkbox"/> | | | |
| | | Inherit rights from the parent roles? | <input checked="" type="checkbox"/> | | | |
| | | Can initiate streaming replication and backups? | <input type="checkbox"/> | | | |

Imagen 10: 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

```

1 CREATE ROLE "usrTest3" WITH
2     LOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     CREATEROLE
6     INHERIT
7     NOREPLICATION
8     CONNECTION LIMIT -1
9     PASSWORD 'xxxxxx';

```

Imagen 11: 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.

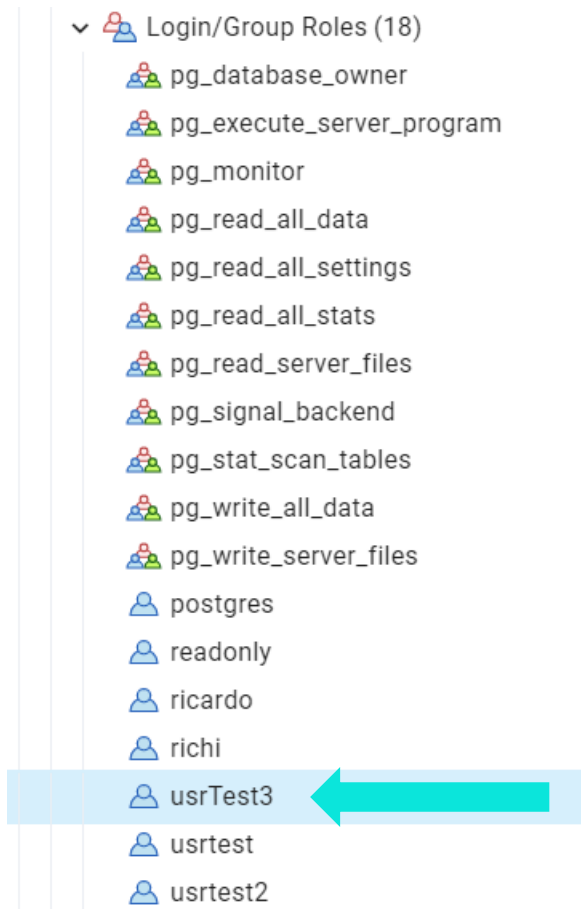


Imagen 12: Rol creado mediante pgAdmin

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

7.- 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).

```
postgres=# REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE
postgres=# REVOKE ALL ON DATABASE pruebas FROM PUBLIC;
REVOKE
postgres=#
```

Imagen 13: 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

8.- 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 4: 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.

```
postgres=# CREATE USER usrtest2 WITH PASSWORD 'usrTest2';
CREATE ROLE
postgres=# \du+
```

| Nombre de rol | Atributos | Miembro de | Descripción |
|---------------|--|------------|-------------|
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | { } | |
| ricardo | | { } | |
| richi | | { } | |
| usrtest | | { } | |
| usrtest2 | | { } | |

Imagen 14: 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 desplegar 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.

```
postgres=# CREATE ROLE readonly;
CREATE ROLE
postgres=# \du+
```

| Lista de roles | | | |
|----------------|--|------------|-------------|
| Nombre de rol | Atributos | Miembro de | Descripción |
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | { } | |
| readonly | No puede conectarse | { } | |
| ricardo | | { } | |
| richi | | { } | |
| usrtest | | { } | |
| usrtest2 | | { } | |

Imagen 15: 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.

9.- 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;


```
postgres=# \du+
```

| Lista de roles | | | |
|----------------|--|------------|-------------|
| Nombre de rol | Atributos | Miembro de | Descripción |
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {} | |
| readonly | No puede conectarse | {} | |
| ricardo | | {} | |
| richi | | {} | |
| usrtest | | {} | |
| usrtest2 | | {} | |

Imagen 16: Error de Login

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

```
postgres=# ALTER ROLE readonly WITH LOGIN;
ALTER ROLE
postgres=# \du+
```

| Lista de roles | | | |
|----------------|--|------------|-------------|
| Nombre de rol | Atributos | Miembro de | Descripción |
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {} | |
| readonly | | {} | |
| ricardo | | {} | |
| richi | | {} | |
| usrtest | | {} | |
| usrtest2 | | {} | |

Imagen 17: 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.

10.- 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)

```
postgres=# ALTER ROLE usrtest WITH CREATEROLE;
ALTER ROLE
```

Imagen 18: 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.

```
postgres=# ALTER ROLE usrtest2 WITH CREATEROLE;
ALTER ROLE
postgres=# GRANT readonly TO usrtest2;
GRANT ROLE
postgres=# \du+
```

| Nombre de rol | Lista de roles Atributos | Miembro de | Descripción |
|---------------|--|------------|-------------|
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {} | |
| readonly | | {} | |
| ricardo | | {} | |
| richi | | {} | |
| usrtest | Crear rol | {readonly} | |
| usrtest2 | Crear rol | {readonly} | |

Imagen 19: 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 es miembro del rol *readonly*.

```
postgres=# REVOKE readonly FROM usrtest;
REVOKE ROLE
postgres=# \du;
```

| Nombre de rol | Lista de roles Atributos | Miembro de | Descripción |
|---------------|--|------------|-------------|
| postgres | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {} | |
| readonly | | {} | |
| ricardo | | {} | |
| richi | | {} | |
| usrtest | Crear rol | {} | |
| usrtest2 | Crear rol | {readonly} | |

Imagen 20: Revocación de permisos

12.- 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



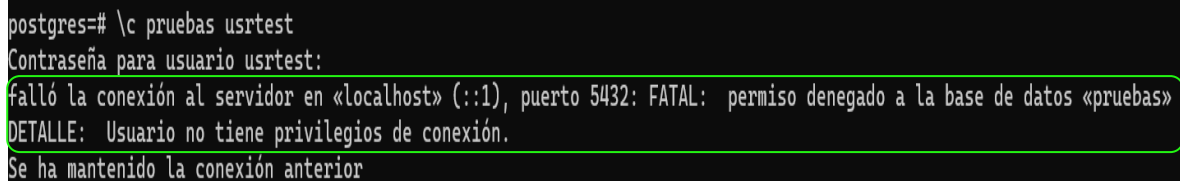
```
postgres=# \c examdep1
Ahora está conectado a la base de datos «examdep1» con el usuario «postgres».
examdep1=#
```

Imagen 21: 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



```
postgres=# \c pruebas usrtest
Contraseña para usuario usrtest:
falló la conexión al servidor en «localhost» (::1), puerto 5432: FATAL:  permiso denegado a la base de datos «pruebas»
DETALLE:  Usuario no tiene privilegios de conexión.
Se ha mantenido la conexión anterior
```

Imagen 22: 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).

```
postgres=# \c pruebas usrtest2
Contraseña para usuario usrtest2:
Ahora está conectado a la base de datos «pruebas» con el usuario «usrtest2».
pruebas=>
```

Imagen 23: 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 *postgres* (Ver imagen 18).

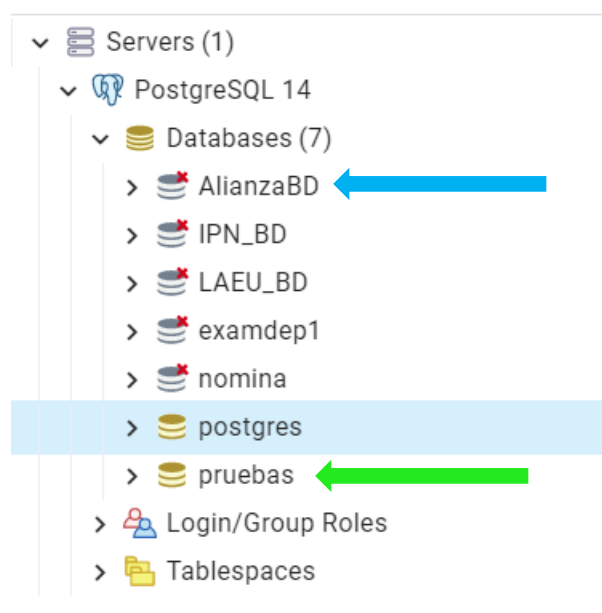


Imagen 24: Listado BD en pgAdmin

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.

13.- 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

```
1 CREATE ROLE readonly2;  
2  
3  
4 GRANT CONNECT ON DATABASE pruebas TO readonly2;  
5  
6  
7 GRANT USAGE ON SCHEMA pruebas_sc TO readonly2;  
8  
9  
10 GRANT SELECT ON ALL TABLES IN SCHEMA pruebas_sc TO readonly2;  
11  
12
```

Imagen 25: 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 específico.

14.- 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.

15.- Glosario

Tabla 5: Glosario

| Término | Descripción |
|------------------|---|
| SQL Shell (Psql) | Es el frontend de la línea de comandos para PostgreSQL |
| pgAdmin 4 | Esta aplicación sirve para gestionar el gestor de BD de PostgreSQL |
| Usuario | Un usuario es todo aquel que tenga contacto con el sistema de bases de datos. |
| Rol | Simplifican el proceso de gestión de privilegios, ya que se pueden otorgar privilegios a un rol y luego otorgar el rol a usuarios |
| PostgreSQL | Es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL |
| Esquema | Un esquema es un conjunto de objetos creado dentro de la misma base de datos para organizar la información mediante tablas, procedimientos, índices, funciones, etc. |
| Base de datos | Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen. |

16.- Bibliografía

Tabla 6: Bibliografía

| Referencias | Título |
|---|--|
| Perdiz, Á. (2020b, junio 1). Gestionar permisos en PostgreSQL. Álvaro Perdiz. https://alvaroperdiz.com/bases-de-datos/postgresql/como-gestionar-permisos/ | Gestionar permisos en PostgreSQL |
| 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/ | Cambiar entre bases de datos en PostgreSQL |
| 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/ | Administración de usuarios y roles de PostgreSQL |

| | |
|---|--|
| - , 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/ | Cómo 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 | PostgreSQL |
| | |
| | |