Cristóbal Suárez Abad

Administración de sistemas gestores de bases de datos  - 2º ASIR

Caso práctico 1 - PostgreSQL

Índice

[***Evidencias*** 3](#_Toc215162001)

[***Enlaces:*** 4](#_Toc215162002)

[**Actividad 1 – Creación de usuarios y roles** 7](#_Toc215162003)

[**Actividad 2 - Creación de vistas personalizadas** 15](#_Toc215162004)

[**Creación de tablas y datos base** 15](#_Toc215162005)

[**Asignación de permisos a usuarios** 21](#_Toc215162006)

[**Ampliación** 25](#_Toc215162007)

[**Actividad 3 – Sinónimos y alias** 26](#_Toc215162008)

[**Creación de sinónimos simulados** 26](#_Toc215162009)

[**Permisos y pruebas de acceso** 27](#_Toc215162010)

[**Actividad 4 – Gestión de privilegios (criterios d–g)** 30](#_Toc215162011)

[**Agrupación de privilegios y rol de solo lectura** 32](#_Toc215162012)

[**Privilegios sobre esquemas** 40](#_Toc215162013)

[**Auditoría final de roles y privilegios** 42](#_Toc215162014)

[**Actividad 5 – Seguridad y cumplimiento** 44](#_Toc215162015)

[**Configuración de políticas de seguridad** 44](#_Toc215162016)

[**Auditoría de accesos** 45](#_Toc215162017)

[Comprobación de seguridad 47](#_Toc215162018)

### ***Evidencias***

* *Será necesario añadir las sentencias ejecutadas así como consultas de comprobación.*

*La empresa TechData S.L., dedicada a la venta y soporte de equipos informáticos, ha decidido implantar una base de datos en PostgreSQL para gestionar la información de clientes y pedidos.*

*Como administrador de bases de datos, tu tarea consiste en configurar los usuarios, roles y privilegios del sistema, creando vistas personalizadas y aplicando medidas de seguridad y control de acceso.*

*El objetivo es que cada perfil (administrador de ventas, empleado y auditor) acceda solo a la información necesaria, garantizando la protección de datos y el principio de mínimo privilegio.*

### ***Enlaces:***

* **CREATE USER:**

[**https://www.postgresql.org/docs/current/sql-createuser.html**](https://www.postgresql.org/docs/current/sql-createuser.html)

**CREATE USER *name* [ [ WITH ] *option* [ ... ] ]**

**where *option* can be:**

**SUPERUSER | NOSUPERUSER**

**| CREATEDB | NOCREATEDB**

**| CREATEROLE | NOCREATEROLE**

**| INHERIT | NOINHERIT**

**| LOGIN | NOLOGIN**

**| REPLICATION | NOREPLICATION**

**| BYPASSRLS | NOBYPASSRLS**

**| CONNECTION LIMIT *connlimit***

**| [ ENCRYPTED ] PASSWORD '*password*' | PASSWORD NULL**

**| VALID UNTIL '*timestamp*'**

**| IN ROLE *role\_name* [, ...]**

**| IN GROUP *role\_name* [, ...]**

**| ROLE *role\_name* [, ...]**

**| ADMIN *role\_name* [, ...]**

**| USER *role\_name* [, ...]**

**| SYSID *uid***

CREATE USER is now an alias for [CREATE ROLE](https://www.postgresql.org/docs/current/sql-createrole.html). The only difference is that when the command is spelled CREATE USER, LOGIN is assumed by default, whereas NOLOGIN is assumed when the command is spelled CREATE ROLE.

* **CREATE ROLE:**

[**https://www.postgresql.org/docs/current/sql-createrole.html**](https://www.postgresql.org/docs/current/sql-createrole.html)

**CREATE ROLE *name* [ [ WITH ] *option* [ ... ] ]**

**where *option* can be:**

**SUPERUSER | NOSUPERUSER**

**| CREATEDB | NOCREATEDB**

**| CREATEROLE | NOCREATEROLE**

**| INHERIT | NOINHERIT**

**| LOGIN | NOLOGIN**

**| REPLICATION | NOREPLICATION**

**| BYPASSRLS | NOBYPASSRLS**

**| CONNECTION LIMIT *connlimit***

**| [ ENCRYPTED ] PASSWORD '*password*' | PASSWORD NULL**

**| VALID UNTIL '*timestamp*'**

**| IN ROLE *role\_name* [, ...]**

**| ROLE *role\_name* [, ...]**

**| ADMIN *role\_name* [, ...]**

**| SYSID *uid***

**CREATE VIEW:**

[**https://www.postgresql.org/docs/current/sql-createview.html**](https://www.postgresql.org/docs/current/sql-createview.html)

**CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ] [ RECURSIVE ] VIEW *name* [ ( *column\_name* [, ...] ) ]**

**[ WITH ( *view\_option\_name* [= *view\_option\_value*] [, ... ] ) ]**

**AS *query***

**[ WITH [ CASCADED | LOCAL ] CHECK OPTION ]**

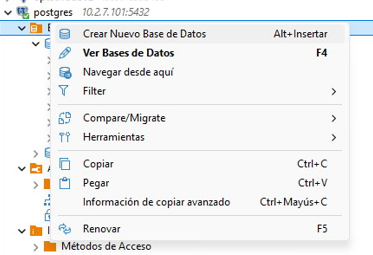
## **Actividad 1 – Creación de usuarios y roles**

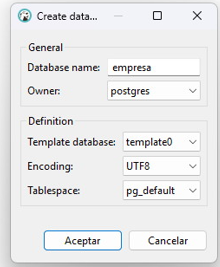
1.- Crea una base de datos llamada empresa.

Podemos crearlo desde el terminal:

**CREATE DATABASE empresa;**

O desde DBeaver:





En DBeaver, una vez que lo creamos con la actual conexión recomiendo crear una nueva conexión solo para la nueva base de datos, para evitar confusiones.

2.- Crea tres usuarios con las opciones definidas:

* admin\_ventas
  + El usuario puede iniciar sesión con la contraseña ‘Audit$2025’
  + Puede crear bases de datos
  + Puede crear y gestionar roles
  + Hereda privilegios de roles asignados
  + Su cuenta expira el 31/12/2026

***CREATE USER admin\_ventas***

***WITH PASSWORD 'Audit$2025'***

***CREATEDB***

***CREATEROLE***

***INHERIT***

***VALID UNTIL '2026-12-31';***

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* empleado\_ventas
  + El usuario puede iniciar sesión con la contraseña ‘Empleado#2025’
  + Tiene un límite de 3 conexiones simultáneas
  + No puede crear roles ni bases de datos

***CREATE USER empleado\_ventas***

***WITH PASSWORD 'Empleado#2025'***

***NOCREATEDB***

***NOCREATEROLE***

***CONNECTION LIMIT 3;***

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* auditor
  + El usuario puede iniciar sesión con la contraseña ‘Audit#2025’
  + No hereda permisos de otros roles
  + Solo puede tener una sesión activa

***CREATE USER auditor***

***WITH PASSWORD 'Audit#2025'***

***NOINHERIT***

***CONNECTION LIMIT 1;***

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

3.- Crea un rol llamado ventas\_grupo.

* No puede iniciar sesión
* Puede heredar privilegios
* No puede crear bases de datos ni roles
* No es superusuario ni tiene permisos de replicación
* No puede omitir políticas de seguridad por filas

***CREATE ROLE ventas\_grupo***

***WITH NOLOGIN***

***INHERIT***

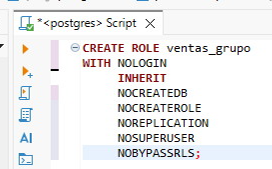
***NOCREATEDB***

***NOCREATEROLE***

***NOREPLICATION***

***NOSUPERUSER***

***NOBYPASSRLS;***



4.- Crea un rol llamado ventas\_acceso

* Puede **iniciar sesión**
* Tiene una **contraseña cifrada** 'Ventas#2025'
* Puede realizar hasta **10 conexiones simultáneas**
* **Hereda privilegios** de otros roles
* No puede crear bases de datos ni roles
* No es superusuario ni tiene permisos de replicación
* No puede omitir políticas de seguridad

***CREATE ROLE ventas\_acceso***

***WITH LOGIN***

***PASSWORD 'Ventas#2025'***

***CONNECTION LIMIT 10***

***INHERIT***

***NOCREATEDB***

***NOCREATEROLE***

***NOREPLICATION***

***NOSUPERUSER***

***NOBYPASSRLS;***

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

5.- Asocia empleado\_ventas y admin\_ventas al rol ventas\_grupo. admin\_ventas tendrá permisos de revocación y asignación al role, empleado\_ventas no.

- Asignación para empleado\_ventas (sin capacidad de administrar el rol)

**GRANT ventas\_grupo TO empleado\_ventas;**

- Asignación para admin\_ventas (con capacidad de administrar el rol)

**GRANT ventas\_grupo TO admin\_ventas WITH ADMIN OPTION;**

**Texto

El contenido generado por IA puede ser incorrecto.**

6.- Crea una tabla con el usuario empleado\_ventas, y luego intenta eliminar el usuario. ¿Qué ocurre? ¿Cómo podrías eliminar el usuario? ¿Qué consecuencias tendría?

Previamente, desde un usuario administrador debemos darle permisos de creación de tablas en esquema public al usuario, porque si no, no podrá crear la tabla:

**GRANT CREATE ON SCHEMA public TO empleado\_ventas;**

Texto

El contenido generado por IA puede ser incorrecto.

Luego el usuario podrá crear la tabla. En este caso la creamos desde una sesión de DBeaver del empleado\_ventas.

***CREATE TABLE public.menu (***

***id varchar NOT NULL,***

***plato varchar NULL,***

***precio varchar NULL,***

***CONSTRAINT restaurante\_pk PRIMARY KEY (id)***

***);***

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Ahora intentamos borrar al usuario desde una cuenta de administrador.

**DROP ROLE empleado\_ventas;**

Texto

El contenido generado por IA puede ser incorrecto.

No podemos, porque hay algunos objetos que dependen de él. Hay que conectarse a todas las bases de datos donde tenga algo el usuario y luego ejecutar los siguientes comandos para quitar cualquier privilegio:

**REVOKE ALL ON ALL TABLES IN SCHEMA public FROM empleado\_ventas;**

**REVOKE ALL ON ALL SEQUENCES IN SCHEMA public FROM empleado\_ventas;**

**REVOKE ALL ON ALL FUNCTIONS IN SCHEMA public FROM empleado\_ventas;**

**REVOKE ALL ON SCHEMA public FROM empleado\_ventas;**

Ahora eliminamos al usuario

**DROP USER empleado\_ventas;**

En este caso lo hemos ejecutado tanto en “**postgres**” como en “**empresa**”.

Texto

El contenido generado por IA puede ser incorrecto.

Consecuencias:

No podemos “logearnos” con sus credenciales, porque ya no existe.

Pero la tabla que creó sigue existiendo: Nos conectamos desde la cuenta de admin\_ventas

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## **Actividad 2 - Creación de vistas personalizadas**

### **Creación de tablas y datos base**

1. Crea una base de datos para el área de ventas llamada ventas\_db

**CREATE DATABASE ventas\_db;**

Y nos conectamos a ella: **\c ventas\_db;**

Texto

El contenido generado por IA puede ser incorrecto.

1. Crea las tablas e inserta algunos registros de ejemplo:
   * clientes (id, nombre, dni, telefono, email, saldo )

***CREATE TABLE clientes (***

***id SERIAL PRIMARY KEY,***

***nombre VARCHAR(100) NOT NULL,***

***dni VARCHAR(15) UNIQUE,***

***telefono VARCHAR(20),***

***email VARCHAR(100),***

***saldo NUMERIC(10, 2) DEFAULT 0.00***

***);***

Texto

El contenido generado por IA puede ser incorrecto.

* + pedidos (id, id\_cliente, fecha, total, estado )

***CREATE TABLE pedidos (***

***id SERIAL PRIMARY KEY,***

***id\_cliente INTEGER REFERENCES clientes(id),***

***fecha DATE DEFAULT CURRENT\_DATE,***

***total NUMERIC(10, 2) NOT NULL,***

***estado VARCHAR(50)***

***);***

Texto

El contenido generado por IA puede ser incorrecto.

1. Inserta algunos registros de ejemplo:

-- Inserción en clientes

***INSERT INTO clientes (nombre, dni, telefono, email, saldo) VALUES***

***('Perro Sanchez', '12345678A', '600111222', 'perrete@psoe.com', -9999999.00),***

***('Donaldo Trumpino', '98765432B', '600333444', 'trump@usa.com', 5000000.00),***

***('Vladimiro Putinino', '11223344C', '600555666', 'ervladi@ruski.com', 1.00);***

Texto

El contenido generado por IA puede ser incorrecto.

-- Inserción en pedidos

***INSERT INTO pedidos (id\_cliente, total, estado) VALUES***

***(1, 150.00, 'Entregado'),***

***(1, 350.50, 'En Proceso'),***

***(2, 75.25, 'Pendiente'),***

***(3, 1200.00, 'Entregado');***

Texto

El contenido generado por IA puede ser incorrecto.

1. Verifica el contenido:

**SELECT \* FROM clientes;**

**SELECT \* FROM pedidos;**

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.

**Creación de vistas personalizadas**

1. Los administradores deben tener acceso total a los datos de clientes y pedidos, con el número de pedidos y total de todos los pedidos

***CREATE VIEW vista\_admin\_ventas AS***

***SELECT***

***c.id AS cliente\_id,***

***c.nombre,***

***c.dni,***

***c.telefono,***

***c.email,***

***c.saldo,***

***COUNT(p.id) AS numero\_pedidos,***

***COALESCE(SUM(p.total), 0) AS total\_gastado***

***FROM***

***clientes c***

***LEFT JOIN***

***pedidos p ON c.id = p.id\_cliente***

***GROUP BY***

***c.id, c.nombre, c.dni, c.telefono, c.email, c.saldo;***

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. Los empleados solo deben ver información de contacto y saldo, sin DNI ni email.

***CREATE VIEW vista\_empleado\_ventas AS***

***SELECT***

***id,***

***nombre,***

***telefono,***

***saldo***

***FROM***

***clientes;***

Texto

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

1. El auditor puede consultar datos, pero sin información personal identificable.

***CREATE VIEW vista\_auditor AS***

***SELECT***

***c.id AS cliente\_id,***

***c.saldo,***

***p.id AS pedido\_id,***

***p.fecha,***

***p.total,***

***p.estado***

***FROM***

***clientes c***

***LEFT JOIN***

***pedidos p ON c.id = p.id\_cliente;***

Texto

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

### **Asignación de permisos a usuarios**

1. Concede permisos de lectura sobre las vistas a cada role o usuarios:

Asignar la vista de administración al usuario admin\_ventas

**GRANT SELECT ON vista\_admin\_ventas TO admin\_ventas;**



Asignar la vista de empleados al usuario empleado\_ventas.

**GRANT SELECT ON vista\_empleado\_ventas TO empleado\_ventas;**



Asignar la vista de auditoría al usuario auditor

**GRANT SELECT ON vista\_auditor TO auditor;**



1. Revoca permisos directos sobre las tablas base, para que solo puedan acceder a través de las vistas

**REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;**



Por si acaso, también podemos usar:

**REVOKE ALL ON clientes FROM ventas\_grupo;**

**REVOKE ALL ON pedidos FROM ventas\_grupo;**

**REVOKE ALL ON clientes FROM auditor;**

**REVOKE ALL ON pedidos FROM auditor;**

1. Comprueba que cada usuario solo puede acceder a su vista correspondiente:

“admin\_ventas”: **SELECT \* from vista\_admin\_ventas;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

“auditor”: **select \* from vista\_auditor;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

“empleado\_ventas”: **select \* from vista\_empleado\_ventas;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

### **Ampliación**

1. Crea una nueva vista vista\_clientes\_negativos que muestre solo clientes con saldo menor que 0.

***CREATE VIEW vista\_clientes\_negativos AS***

***SELECT***

***id,***

***nombre,***

***telefono,***

***saldo***

***FROM***

***clientes***

***WHERE***

***saldo < 0;***

Texto

El contenido generado por IA puede ser incorrecto.

1. Asigna esta vista al rol ventas\_acceso para que todos los usuarios de ventas puedan consultarla.

**GRANT SELECT ON vista\_clientes\_negativos TO ventas\_acceso;**



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## **Actividad 3 – Sinónimos y alias**

### **Creación de sinónimos simulados**

*Recuerda:* PostgreSQL no tiene el comando CREATE SYNONYM (como Oracle o SQL Server).

* Crea una vista simple que actúe como alias vista\_clientes\_admin de la tabla clientes:

***CREATE VIEW vista\_clientes\_admin AS***

***SELECT \****

***FROM public.clientes;***

Texto

El contenido generado por IA puede ser incorrecto.

* Crea otra vista que funcione como sinónimo de la vista vista\_clientes\_admin:

***CREATE VIEW vca\_sinonimo AS***

***SELECT \****

***FROM vista\_clientes\_admin;***

Texto

El contenido generado por IA puede ser incorrecto.

* Crea un alias más complejo para la vista de empleados, que renombre columnas

***CREATE VIEW ventas\_contacto AS***

***SELECT***

***id AS cliente\_id,***

***nombre AS nombre\_cliente,***

***telefono AS contacto\_principal,***

***saldo AS deuda\_o\_credito***

***FROM***

***vista\_empleado\_ventas;***

Texto

El contenido generado por IA puede ser incorrecto.

### **Permisos y pruebas de acceso**

* Concede permisos de lectura sobre los alias a los usuarios o roles que consideres.

**GRANT SELECT ON vista\_clientes\_admin TO admin\_ventas;**

**GRANT SELECT ON vca\_sinonimo TO admin\_ventas;**

**GRANT SELECT ON ventas\_contacto TO ventas\_grupo;**

Texto

El contenido generado por IA puede ser incorrecto.

* Comprueba la diferencia entre un usuario con permisos y sin permisos

Si lo hacemos desde el “admin\_ventas” funciona.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Si lo intentamos desde del “auditor” no funciona.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

## **Actividad 4 – Gestión de privilegios (criterios d–g)**

**Comprobación previa**

* Lista los privilegios actuales sobre todas las tablas y vistas[[1]](#footnote-1):

**\dp \*.\***



Texto

El contenido generado por IA puede ser incorrecto.

* Comprueba también los roles y sus pertenencias:

**\du**

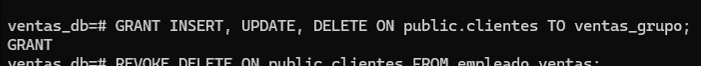
Texto

El contenido generado por IA puede ser incorrecto.

**Asignación y prueba de privilegios**

* Sobre la tabla clientes
  + Concede privilegios de manipulación a ventas\_grupo sobre clientes

**GRANT SELECT, INSERT, UPDATE, DELETE ON public.clientes TO ventas\_grupo;**



* + Da permisos de eliminación **solo** al usuario admin\_ventas:

**GRANT DELETE ON public.clientes TO admin\_ventas;**

Texto

El contenido generado por IA puede ser incorrecto.

* + Revoca explícitamente el permiso de eliminación a empleado\_ventas:

**REVOKE DELETE ON public.clientes TO empleado\_ventas;**

* + Verifica el efecto práctico:

Desde el usuario admin\_ventas:

**DELETE FROM public.clientes**

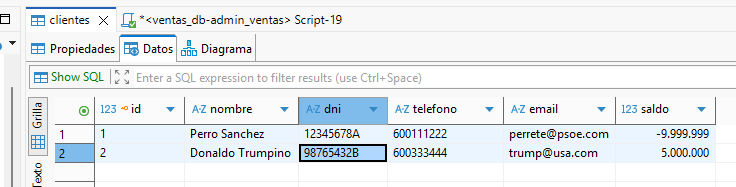
**WHERE id=3;**

En teoría podemos borrarlo, pero no nos deja porque está siendo usada en otra tabla.

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Si desde un usuario con permisos borramos las entradas del cliente con “id” igual a tres en la tabla “pedidos” (**DELETE FROM pedidos WHERE id\_cliente = 3;**), podremos hacer que el comando tenga efecto:



Desde el usuario **empleado\_ventas**:

**DELETE FROM public.clientes**

**WHERE id=3;**

A este no le deja porque no tiene permiso.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

### **Agrupación de privilegios y rol de solo lectura**

* Crea un rol de solo lectura:

**CREATE ROLE solo\_lectura NOLOGIN;**

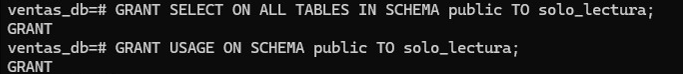


* Concede privilegios de lectura sobre todas las tablas y vistas actuales:

**GRANT SELECT ON ALL TABLES IN SCHEMA public TO solo\_lectura;**

Para poder acceder a los objetos también hay que darle este permiso:

**GRANT USAGE ON SCHEMA public TO solo\_lectura;**



* Haz que los **nuevos objetos creados** también sean legibles por este rol:

**ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO solo\_lectura;**



* Asigna el rol al usuario auditor:

**GRANT solo\_lectura TO auditor;**

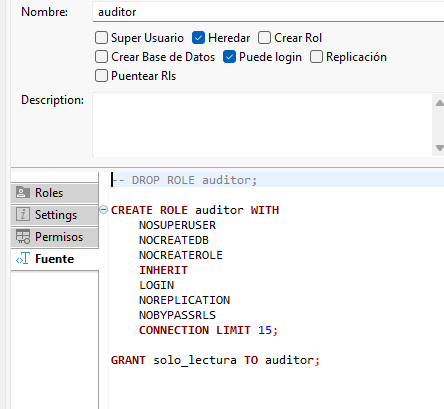


* Comprueba que el usuario auditor hereda los permisos:

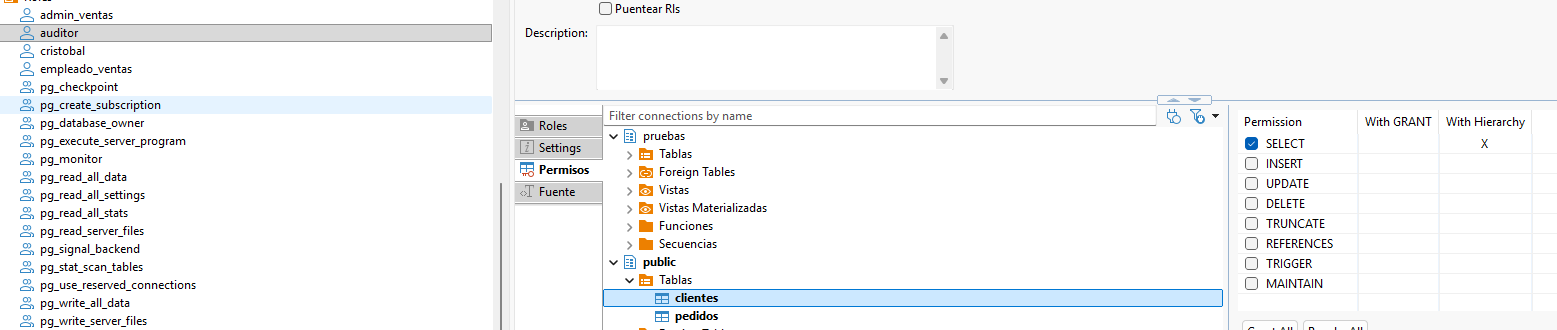
Su configuración modificada para que pueda heredar:

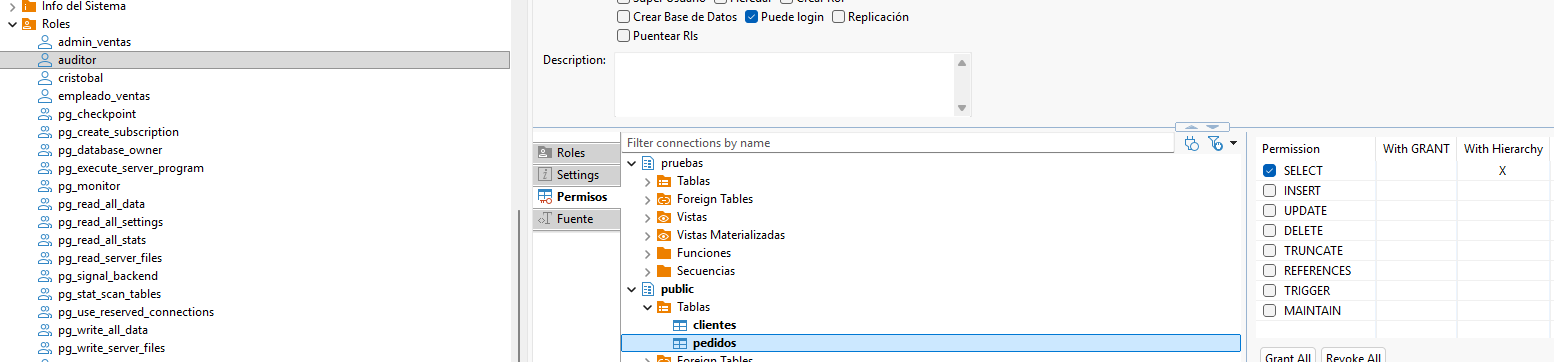
**ALTER ROLE auditor NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN NOREPLICATION NOBYPASSRLS;**



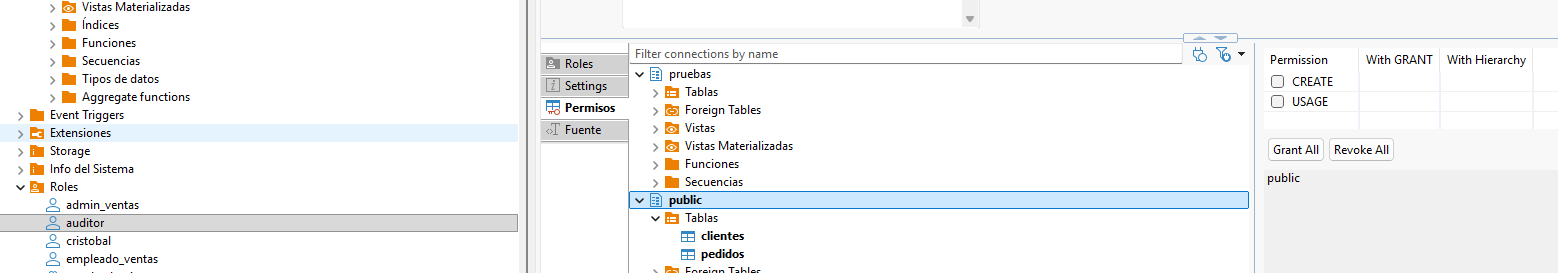


Los hereda en ambas tablas.





Por motivos desconocidos no nos aparece en el esquema la herencia de “**Usage**”:



* Inicia sesión como auditor y prueba:

Desde la sesión del auditor, ejecutamos:

**SELECT \* FROM public.clientes;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

**SELECT \* FROM pedidos;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

**Gestión dinámica de privilegios**

Sobre la tabla clientes:

* Elimina los permisos del grupo de ventas:

**REVOKE ALL ON public.clientes FROM ventas\_grupo;**

Texto

El contenido generado por IA puede ser incorrecto.

* Observa el efecto

Desde el usuario “admin\_ventas”:

**SELECT \* FROM public.clientes;**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.



**No tiene permiso.**

* Añade privilegios de consulta

**GRANT SELECT ON public.clientes TO ventas\_grupo;**



* Comprueba si el permiso vuelve a estar disponible.

**Funciona**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

### **Privilegios sobre esquemas**

* Crea un nuevo esquema llamado pruebas:

**CREATE SCHEMA pruebas;**

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* Concede a ventas\_grupo permiso para usar el esquema, pero no para crear objetos

**GRANT USAGE ON SCHEMA pruebas TO ventas\_grupo;**



* Intenta crear una tabla dentro del esquema con empleado\_ventas y verifica el resultado.

No se puede: No tiene permisos.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* Da permisos para crear

**GRANT CREATE ON SCHEMA pruebas TO ventas\_grupo;**



* Vuelve a probar y observa la diferencia.

Ahora ya se puede.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

### 

### **Auditoría final de roles y privilegios**

* Consulta todos los privilegios otorgados a cada usuario:

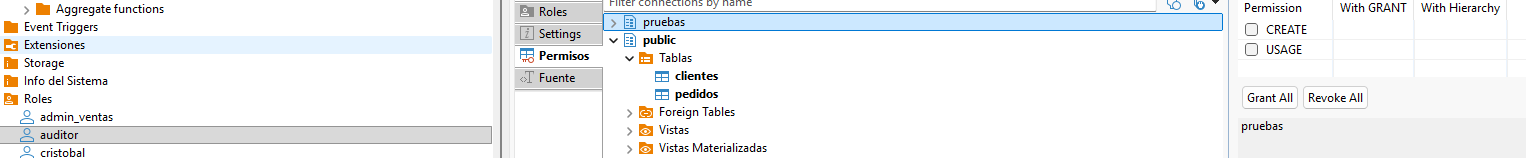
Para esquema “pruebas”:

“admin\_ventas”

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

“auditor”



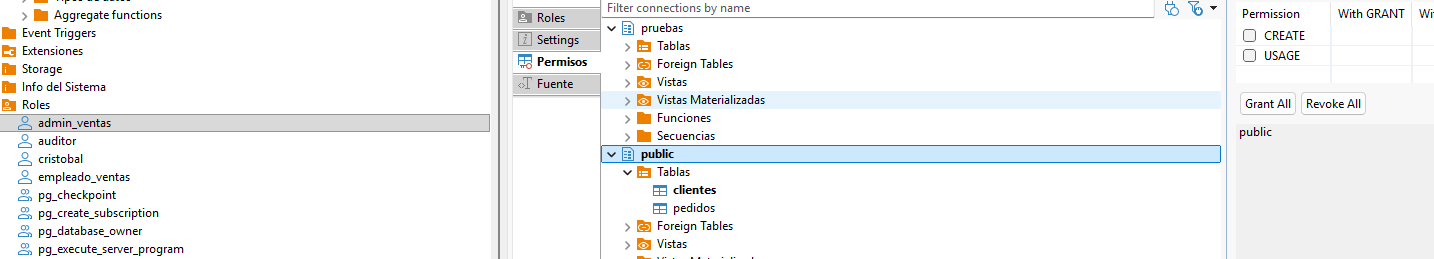
“empleado\_ventas”

Interfaz de usuario gráfica, Aplicación

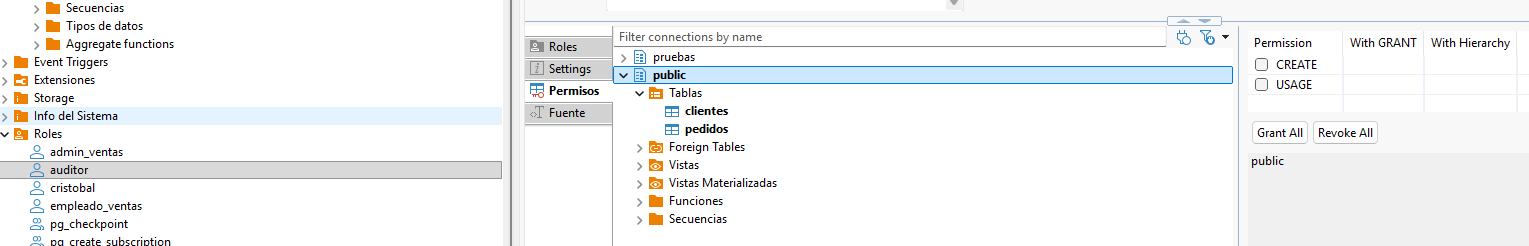
El contenido generado por IA puede ser incorrecto.

Para esquema “public”:

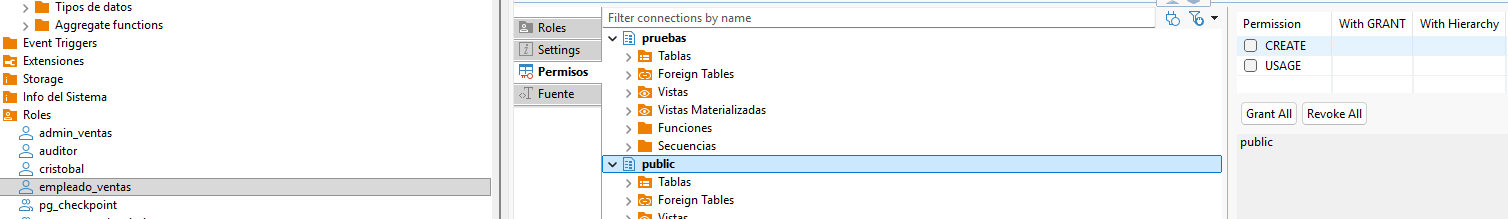
“admin\_ventas”:



“auditor”:



“empleado\_ventas”:



# **Actividad 5 – Seguridad y cumplimiento**

## **Configuración de políticas de seguridad**

* + Activa el registro (logging\_collector = on) y revisa el archivo postgresql.conf.

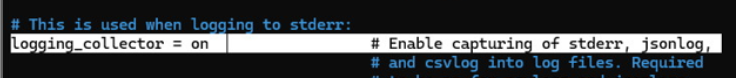
Modificamos el archivo “**postgresql.conf**”. Debemos poner:

**logging\_collector = on**

Y activar también las opciones:

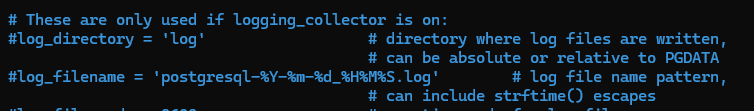
**log\_connections = on**

**log\_disconnections = on**

 Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Algunos compañeros han dicho que han tenido que “descomentar” las siguientes líneas para que les funcione el registro. A mi no me ha hecho falta.



* + Cambia la política de autenticación de md5 a scram-sha-256 en pg\_hba.conf.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

## **Auditoría de accesos**

* + Conéctate con distintos usuarios y revisa los logs.

Para saber dónde están los logs:

Te metes en **postgresql:**

Encuentra el Directorio de Datos (PGDATA): **SHOW data\_directory**;

Verifica el Directorio de Logs: **SHOW log\_directory;**

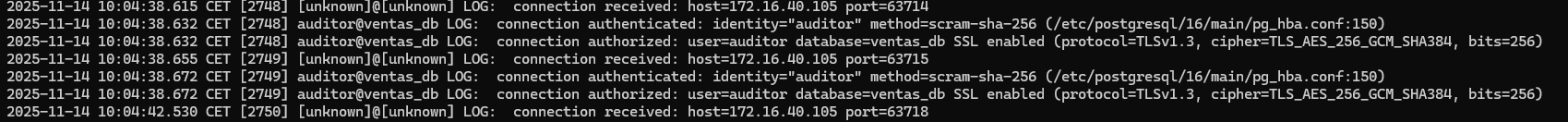
Determina el Nombre del Archivo de Log Actual (si el servidor está en ejecución): **SELECT pg\_current\_logfile();**

Texto

El contenido generado por IA puede ser incorrecto.

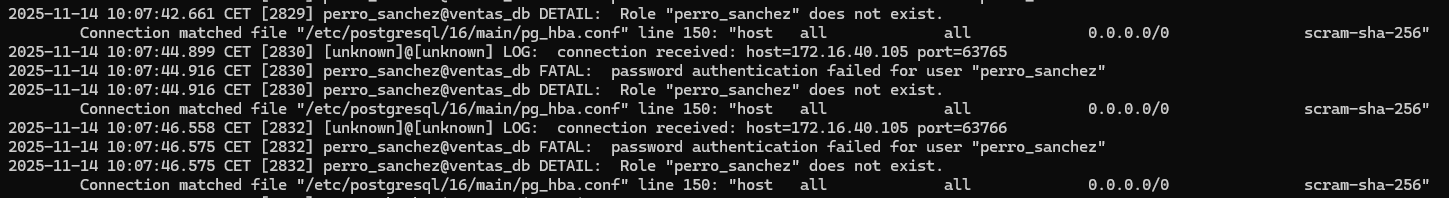
En “/var/lib/postgresql/16/main/log/”:

“Logeo” exitoso:



* + Identifica intentos fallidos de conexión. Explica que ves

“Logeo” fallido:



Me indica hora, IP y nombre del usuario que intenta conectarse, también la conexión configurada en “pg\_hba.conf” que se le aplicaría.

### Comprobación de seguridad

* + Verifica los privilegios de cada usuario (\du y \z).

Texto

El contenido generado por IA puede ser incorrecto.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

* Elabora un informe final indicando qué medidas garantizan la seguridad.
* **Vistas personalizadas**: Cada usuario solo accede a las columnas y datos esenciales para su función, ocultando información sensible (DNI, Email).
* **Revocación de Permisos Directos**: Se fuerza a los usuarios a usar las vistas, asegurando que las políticas de filtrado y exposición de datos se cumplan.
* **Restricción de Conexiones:** Previene el abuso o el uso excesivo de recursos por parte de cuentas individuales.
* **Asignación de Permisos de Manipulación Detallada**: Solo admin\_ventas tiene el poder de eliminar registros, mientras que empleado\_ventas solo puede insertar/actualizar.
* **Rol solo\_lectura**: Centraliza y simplifica la concesión de acceso de solo lectura para la auditoría.
* **Autenticación scram-sha-256**: Mejora la seguridad de las contraseñas almacenadas, haciéndolas menos susceptibles a ataques que la política md5.
* **Registro de Actividad**: Permite detectar y rastrear intentos fallidos, accesos no autorizados y actividades sospechosas, garantizando el cumplimiento.

1. <https://supabase.com/blog/postgres-roles-and-privileges> [↑](#footnote-ref-1)