

TAREA 5. PosgreSQL

Villeda Tlecuitl José Eduardo

August 26, 2024

1 Como crear usuarios y roles en el manejador y cuales propiedades le puedo asignar.

- Un usuario en PostgreSQL es un tipo especial de rol que tiene permiso para iniciar sesión en la base de datos.

Crear un Usuario: Un usuario en PostgreSQL es esencialmente un rol con permiso de inicio de sesión.

CODIGO

CREATE USER nombre_usuario **WITH PASSWORD** 'contraseña';

- Un rol es una entidad dentro de PostgreSQL a la que se le pueden asignar permisos.

Crear un Rol: Los roles en PostgreSQL son entidades a las que puedes asignar permisos (privilegios) en la base de datos. A veces, los roles no tienen permisos de inicio de sesión y solo sirven para agrupar permisos.

CODIGO

CREATE ROLE nombre_rol;

- Si deseas que un rol también funcione como un usuario, puedes agregarle la capacidad de iniciar sesión.

CODIGO

CREATE ROLE nombre_rol **WITH LOGIN PASSWORD** 'contraseña';

1.1 Propiedades de los roles y usuarios.

- **PASSWORD:** Establece la contraseña para los roles con la opción de inicio de sesión.
- **IN ROLE:** Asigna al usuario a uno o más roles.
- **SUPERUSER:** Permite que el rol sea un superusuario, con acceso total al sistema.
- **LOGIN:** Permite al rol iniciar sesión. Es lo que diferencia a un usuario de un rol genérico.
- **DEFAULT PRIVILEGES:** Permite configurar privilegios predeterminados que serán aplicados automáticamente a objetos creados por un rol en el futuro.

2 Que son los privilegios a nivel sistemas y a nivel objetos, ejemplos de cada uno.

Los privilegios son permisos que definen qué acciones puede realizar un usuario o rol dentro del sistema. Estos privilegios se dividen en dos grandes categorías:

- Privilegios a nivel de sistema.
- Privilegios a nivel de objeto.

2.1 Privilegios a nivel de sistema.

Los privilegios a nivel del sistema son autorizaciones que determinan qué acciones un rol o usuario puede ejecutar en toda la instancia de PostgreSQL. Estos permisos impactan la gestión global del sistema, la creación de bases de datos y roles, así como otras operaciones de nivel superior.

- **CREATEROLE:** Permite al rol crear, modificar y eliminar otros roles dentro de la base de datos.
- **NOCREATEDB:** Niega la capacidad de crear bases de datos.
- **LOGIN:** Permite que el rol inicie sesión en la base de datos. Sin este privilegio, el rol no puede conectarse a la base de datos, aunque tenga otros privilegios.
- **CREATEDB:** Permite crear nuevas bases de datos.
- **INHERIT:** Permite que un rol herede privilegios de otros roles.
- **REPLICATION:** Permite que el rol participe en la replicación de bases de datos, que se usa para copiar datos entre servidores.

2.2 Privilegios a nivel de objeto.

Los privilegios a nivel de objeto determinan qué acciones puede llevar a cabo un rol o usuario en objetos específicos dentro de una base de datos. Un objeto puede ser una tabla, vista, secuencia, esquema, función o base de datos, entre otros. Estos permisos permiten limitar o autorizar las acciones que un usuario o rol puede realizar sobre dichos objetos.

- **DELETE:** Permite borrar filas de una tabla.
- **ALTER:** Permite modificar la definición de un objeto, como una tabla, secuencia o función.
- **CREATE:** Permite crear nuevos objetos dentro de un esquema, como tablas o vistas.
- **SELECT:** Permite consultar datos de una tabla, vista o secuencia.
- **SET:** Permite modificar configuraciones de sesión o de base de datos.

3 Como otorgar y quitar privilegios a un usuario.

3.1 Como otorgar privilegios a un usuario.

La sentencia GRANT se utiliza para otorgar privilegio a un usuario.

SINTAXIS:

GRANT privilegio(s) ON objeto TO usuario;

Ejemplo:

Otorgar permisos de EXECUTE sobre una función llamada mi_funcion al usuario usuario1:

GRANT EXECUTE ON FUNCTION mi_funcion() TO usuario1;

3.2 Como quitar privilegios a un usuario

La sentencia REVOKE se utiliza para retirar permisos previamente concedidos a un usuario.

SINTAXIS:

REVOKE privilegio(s) ON objeto FROM usuario;

Ejemplo:

Quitar permisos de EXECUTE sobre una función llamada mi_funcion al usuario usuario1:

REVOKE EXECUTE ON FUNCTION mi_funcion() FROM usuario1;

References

- “Administración de usuarios y roles de PostgreSQL — Amazon Web Services”. Amazon Web Services. Accedido el 26 de agosto de 2024. [En línea]. Disponible: <https://aws.amazon.com/es/blogs/aws-spanish/managing-postgresql-users-and-roles/>
- “Administracion de usuarios y de roles de PostgreSQL”. Cloud, DevOps, Seguridad, Soporte Informatico — Qualoom. Accedido el 26 de agosto de 2024. [En línea]. Disponible: <https://www.qualoom.es/blog/administracion-usuarios-roles-postgresql/>
- “Acerca de los usuarios y los roles de PostgreSQL — Cloud SQL for PostgreSQL — Google Cloud”. Google Cloud. Accedido el 26 de agosto de 2024. [En línea]. Disponible: <https://cloud.google.com/sql/docs/postgres/users?hl=es-419>
- “Configuración de usuarios PostgreSQL”. WakerVall. Accedido el 26 de agosto de 2024. [En línea]. Disponible: <https://wakervall.wordpress.com/2018/02/03/configuracion-de-usuarios-postgresql/>