Tarea 5

Nombre del alumno: Molina Véjar Aarón Gael

Profesor: Ing. Fernando Arreola Franco

Bases de datos

¿Cómo crear usuarios y roles en PostgreSQL?

Usuarios

Para crear un usuario se usa el comando SQL CREATE USER, el cual tiene la siguiente estructura:

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

Donde CREATE USER es el comando principal, **WITH** es una palabra opcional pero que es recomendable escribir para una mejor legibilidad del comando, y **option** puede ser reemplazado por las siguientes palabras reservadas:

```
 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
```

Consideraciones importantes: CREATE USER es un alias de CREATE ROLE, esto significa que ambos comandos poseen un comportamiento funcional idéntico, pero la diferencia radica en la configuración para iniciar sesión (LOGIN) que poseen por defecto (versión 16 de PostgreSQL).

- CREATE USER: Cuando se utiliza este comando, se asume por defecto que el nuevo rol tendrá la capacidad de iniciar sesión (LOGIN).
- CREATE ROLE: Cuando se utiliza este comando, se asume por defecto que el nuevo rol no tendrá la capacidad de iniciar sesión (NOLOGIN).

Roles

Para crear un rol se usa el comando SQL CREATE ROLE, el cual tiene la siguiente estructura:

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

De la misma manera que el comando para crear un usuario, **option** se puede reemplazar por las siguientes palabras reservadas:

```
 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 ROLE agrega un nuevo rol a un clúster de la base de datos, por lo que el rol se almacena a nivel clúster y todas las bases de datos que se encuentren en él pueden acceder a los roles. Un rol es una entidad que puede poseer objetos y privilegios de una base de datos; un rol puede considerarse un "usuario" o un "grupo" dependiendo de cómo se utilice.

Durante la creación de roles, es posible asignar inmediatamente el rol recién creado para que sea miembro de un rol existente, y también asignar roles existentes para que sean miembros del rol recién creado.

Privilegios a nivel sistema y a nivel objetos

Los privilegios que se pueden otorgar en una base de datos dentro de PostrgeSQL se dividen en dos categorías básicas, los privilegios de objeto y los privilegios de rol (también llamados privilegios a nivel sistema).

Privilegios a nivel objetos

Los privilegios en objetos se aplican sobre los siguientes elementos: tabla, columna, vista, tabla foránea, secuencias, bases de datos, contenedor de datos externos, servidor externo, función, procedimiento, lenguaje de procedimientos, objeto grande, parámetro de configuración, esquema, espacio de tabla o tipo.

Ejemplos:

Estructura del comando SQL aplicado en una tabla, una secuencia y una base de datos:

```
GRANT { { SELECT | INSERT | UPDATE | REFERENCES } ( column_name [, ...] )
    [, ...] | ALL [ PRIVILEGES ] ( column_name [, ...] ) }
    ON [ TABLE ] table_name [, ...]
    TO role_specification [, ...] [ WITH GRANT OPTION ]
    [ GRANTED BY role_specification ]

GRANT { { USAGE | SELECT | UPDATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { SEQUENCE sequence_name [, ...]
      | ALL SEQUENCES IN SCHEMA schema_name [, ...] }
    TO role_specification [, ...] [ WITH GRANT OPTION ]
    [ GRANTED BY role_specification ]

GRANT { { CREATE | CONNECT | TEMPORARY | TEMP } [, ...] | ALL [ PRIVILEGES ] }
    ON DATABASE database_name [, ...]
    TO role_specification [, ...] [ WITH GRANT OPTION ]
    [ GRANTED BY role_specification ]
```

Posibles privilegios

```
SELECT
INSERT
UPDATE
DELETE
TRUNCATE
REFERENCES
TRIGGER
CREATE
CONNECT
TEMPORARY
EXECUTE
USAGE
SET
ALTER SYSTEM
TEMP
ALL PRIVILEGES
```

Privilegios a nivel sistema o rol

Estos privilegios se aplican directamente a un rol.

Ejemplos:

Estructura de comando SQL aplicado a un rol

```
GRANT role_name [, ...] TO role_specification [, ...]
    [ WITH { ADMIN | INHERIT | SET } { OPTION | TRUE | FALSE } ]
    [ GRANTED BY role_specification ]

where role_specification can be:

    [ GROUP ] role_name
  | PUBLIC
  | CURRENT_ROLE
  | CURRENT_USER
  | SESSION_USER
```

Posibles privilegios

```
SELECT
INSERT
UPDATE
DELETE
TRUNCATE
REFERENCES
TRIGGER
CREATE
CONNECT
TEMPORARY
EXECUTE
USAGE
SET
ALTER SYSTEM
TEMP
ALL PRIVILEGES
```

¿Cómo otorgar y quitar privilegios a un usuario?

Como se observó anteriormente para asignar un privilegio se utiliza la palabra reservada GRANT, y para quitar un privilegio se utiliza REVOKE, por lo que tenemos lo siguiente:

```
GRANT privilegio ON objeto TO usuario;

REVOKE privilegio ON objeto FROM usuario;
```