

ROLES Y PRIVILEGIOS EN POSTGRESQL

Roles:

Los permisos de acceso en PostgreSQL se manejan bajo un concepto de rol, que se asemeja a un usuario, sin embargo, también puede representar un grupo de usuarios. Un rol es a quien se le asignan los privilegios de acción y acceso sobre ciertos objetos en una base de datos. Los roles de la base de datos están conceptualmente completamente separados de los usuarios del sistema operativo.

Atributos para roles:

- LOGIN
- SUPERUSER
- CREATEROLE
- REPLICATION LOGIN
- PASSWORD



Privilegios:

Cuando se crea un objeto, se le asigna un propietario. El propietario es normalmente el rol que ejecutó la declaración de creación. Para la mayoría de los tipos de objetos, el estado inicial es que solo él puede hacer algo con el objeto. Para permitir que otros roles lo utilicen, se deben otorgar *privilegios*.

Atributos para privilegios:

Privilegio	Abreviatura	Tipos de objetos aplicables
SELECT	r(" Leer ")	LARGE OBJECT, SEQUENCE, TABLE(Y de tabla de objetos), columna de la tabla
INSERT	a(" Añadir ")	TABLE, columna de la tabla
UPDATE	w(" Escribir ")	LARGE OBJECT, SEQUENCE, TABLE, Columna de la tabla
DELETE	d	TABLE
TRUNCATE	D	TABLE
REFERENCES	x	TABLE, columna de la tabla
TRIGGER	t	TABLE
CREATE	C	DATABASE, SCHEMA, TABLESPACE
CONNECT	c	DATABASE
TEMPORARY	T	DATABASE
EXECUTE	X	FUNCTION, PROCEDURE
USAGE	U	DOMAIN, FOREIGN DATA WRAPPER, FOREIGN SERVER, LANGUAGE, SCHEMA, SEQUENCE, TYPE

DIFERENCIA USUARIO – ROL Y MODELO JERÁRQUICO - RED

Diferencia entre usuario y rol

USUARIO: Todas aquellas personas que tienen acceso al sistema y son miembros de uno a más grupos.

ROL: Son los permisos que se van a tener habilitados al momento de operar con el sistema y tiene acceso a objetos de bases de datos.

En versiones modernas de PostgreSQL, ambos conceptos se han fusionado, un rol puede tener la capacidad de iniciar sesión.

Diferencia entre modelo jerárquico y de red

Jerárquico: La estructura de sus datos es escalonada (nodo padre y nodos hijos), se asemeja al árbol. Se pueden representar relaciones de uno a uno y de uno a muchos.

Red: En este modelo se puede tener más de un padre, por lo que es más flexible, se tienen registros en forma de grafo. No hay redundancia en la información, a través de la incorporación de un conector.

Referencias:

<https://www.postgresql.org/docs/current/database-roles.html>

<https://www.postgresql.org/docs/current/ddl-priv.html>

<https://medium.com/@dgzraul.web/gesti%C3%B3n-de-roles-y-privilegios-en-postgresql-12-dc6897445a29>