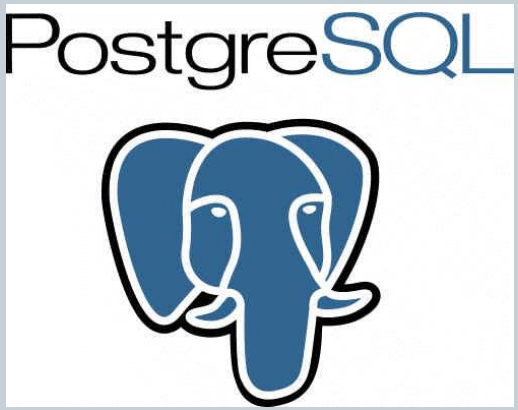
CURSO DE FUNDAMENTOS Y ADMINISTRACION DE DBMS POSTGRES

UBUNTU 10.4



INSTRUCTOR: RUDY SALVATIERRA RODRIGUEZ

POSTGRES:ADMINISTRACION DE ROLES

- Después de ingresar al servidor de postgres creamos el rol llamado grupo1 con password grupo1. ,de la siguiente manera

  CREATE ROLE grupo1 WITH ENCRYPTED PASSWORD ‘grupo1’;

- Luego creamos 2 usuarios llamados usr1 y usr2 con sus password

  CREATE USER usr1 WITH ENCRYPTED PASSWORD ‘usr1’;

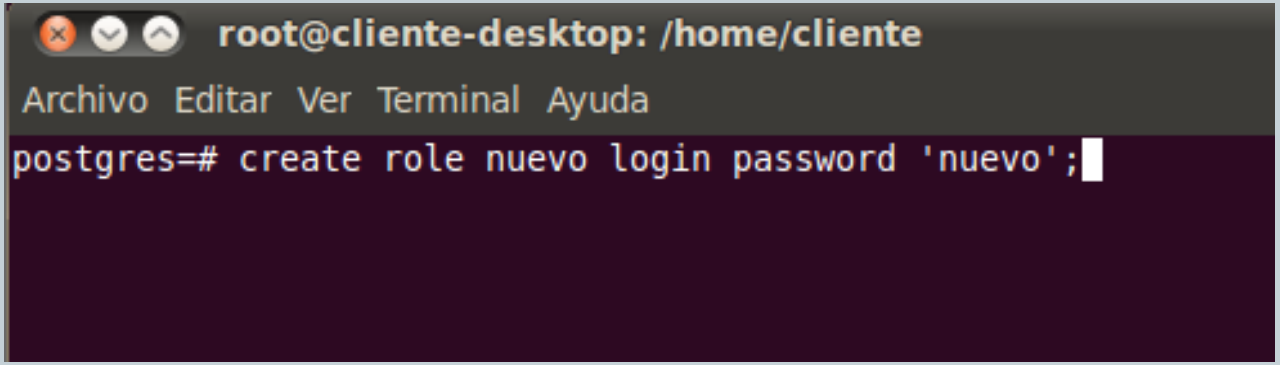
  CREATE USER usr2 WITH ENCRYPTED PASSWORD ‘usr2’;

POSTGRES:ADMINISTRACION DE USUARIOS

- Luego asignamos los usuarios al rol grupo1 de la siguiente manera:

GRANT ROLE grupo1 to usr1,usr2;

- Si quisiéramos crear una rol que se pueda conectar con un password se lo realiza de la siguiente manera.



POSTGRES:ADMINISTRACION DE USUARIOS

- En primer lugar, los roles de los miembros que tienen el atributo INHERIT tienen automáticamente el uso de los privilegios de los roles a los que pertenecen. Como ejemplo, supongamos que hemos hecho:
- Si quisiéramos crear una rol que se pueda conectar con un password se lo realiza de la siguiente manera.

```
CREATE ROLE joe LOGIN INHERIT;
CREATE ROLE admin NOINHERIT;
CREATE ROLE wheel NOINHERIT;
GRANT admin TO joe;
GRANT wheel TO admin;
```

POSTGRES:ADMINISTRACION DE USUARIOS

- Los miembros de un rol puede usar los privilegios del rol de grupo de dos maneras.
- En segundo lugar, cada miembro de un grupo puede hacer de forma explícita SET ROLE para "convertirse" temporalmente en el rol de grupo.
- En este estado, el período de base de datos tiene acceso a los privilegios de la función del grupo por encima del rol de acceso original, y cualquier objeto de base de datos creada se considera propiedad del rol de grupo y no de la función de acceso.

```
SET ROLE admin;
```

POSTGRES:ADMINISTRACION DE USUARIOS

- Para volver al estado de privilegio original puede ser restaurado con cualquiera de:

```
SET ROLE joe;
SET ROLE NONE;
RESET ROLE;
```

El comando SET ROLE siempre permite seleccionar cualquier role que la función original de inicio de sesión es directa o indirectamente de un miembro. Así, en el ejemplo anterior, no es necesario convertirse en administrador antes de convertirse en wheel.

POSTGRES:ADMINISTRACION DE USUARIOS

- En el estándar SQL, hay una distinción clara entre los usuarios y roles, y los usuarios no heredan automáticamente los privilegios, mientras que los roles hacen que este comportamiento se puede obtener utilizando las funciones de SQL como el atributo INHERIT.
- El papel de los atributos de LOGIN, SUPERUSER, CREATEDB, y CREATEROLE pueden considerarse como privilegios especiales, pero nunca se heredan como los privilegios de los objetos ordinarios de la base de datos.
- Se debe si o si establecer SET ROLE a un rol específico con uno de estos atributos a fin de hacer uso del atributo.

POSTGRES:ADMINISTRACION DE USUARIOS

- Continuando con el ejemplo anterior, podríamos optar por otorgar CREATEDB y CREATEROLE a la función de administrador. Luego de una sesión de conexión como Joe no tendríamos estos privilegios de inmediato, sólo después de hacer SET ROLE admin.

POSTGRES:ADMINISTRACION DE USUARIOS

- El manejo de roles en PostgreSQL permite diferentes configuraciones, entre ellas estan:
- **SUPERUSER/NOSUPERUSER**. Súper usuario, privilegios para crear bases de datos y usuarios.
- **CREATEDB/NOCREATEDB**. Permite crear bases de datos.
- **CREATEROLE/NOCREATEROLE**. Permite crear roles.
- **CREATEUSER/NOCREATEUSER**. Permite crear usuarios.
- **LOGIN/NOLOGIN**. Este atributo hace la diferencia entre un rol y usuario. Ya que el usuario tiene permisos para acceder a la base de datos a traves de un cliente.
- **PASSWORD**. Permite alterar la contraseña.
- **VALID UNTIL**. Expiración de usuarios.

POSTGRES:ADMINISTRACION DE USUARIOS

- Si quisiéramos ver los privilegios de conexión de un rol hacia una base de datos podemos verlo con ejecutando la siguiente función

```
SELECT has_database_privilege(‘rol', ‘bd', 'connect');
```

- Todas estas funciones retornan un valor boolean (t, f) indicando si el usuario tiene o no el privilegio indicado sobre el objeto. Para averiguar si todos los usuarios pueden conectarse a la base de datos

POSTGRES:ADMINISTRACION DE USUARIOS

- Si quisiéramos ver los privilegios del rol usrprueba sobre la base datos se lo realiza de la siguiente manera.

```
root@cliente-desktop: /home/cliente
Archivo  Editar  Ver  Terminal  Ayuda
postgres=# SELECT has_database_privilege('usrprueba', 'prueba', 'connect');
```

Nos mostraría los siguiente

```
root@cliente-desktop: /home/cliente
Archivo  Editar  Ver  Terminal  Ayuda
postgres=# SELECT has_database_privilege('usrprueba', 'prueba', 'connect');
 has_database_privilege
------------------------
 t
(1 fila)
```

POSTGRES:ADMINISTRACION DE USUARIOS

- En un clúster con varias bases de datos podemos limitar el acceso quitándole el privilegio a la base de datos prueba y otorgándoselo a nuestros usuarios.

  usrprueba@prueba=> REVOKE ALL ON DATABASE prueba FROM usrprueba ;

- Y asignamos a nuestro usuario

  usrprueba@prueba => GRANT CONNECT ON DATABASE prueba TO userprueba;