

PRACTICAS ROLES:

1) Desde psql:

- a. Crear una nueva BD.
- b. Crear un role administrador que sea: superuser, pueda hacer login, tenga una password asignada: 'miclave' y sea válido hasta que acabe el 2023.
- c. Una vez creado el role conectarse a psql con administrador.
- d. Crear un nuevo role con administrador: role andrea, con login y password y que se añada al role de administrador.
- e. Probar si funciona
- f. Crear otro role: carlos que no puede hacer login y se añade a administrador.
- g. Probar a conectar con carlos.

2) Conectar con el role administrador y:

- a. Probar create role consulta.
- b. Crear otro role: jonathan que tiene login y una password, indicar que no tiene permisos de crear bases de datos y le añadimos al role de consulta.
- c. Conectar con jonathan y comprobar si puede o no crear BD.

3) Probar a crear roles con pgAdmin. Se puede conectar el servidor con este nuevo usuario en pgAdmin.

*En las siguientes prácticas como hay que estar cambiando de usuario puede ser útil: **select user**, para saber con que usuario estamos conectados.*

4) Disponemos del fichero: practicas/base-taller-012-roles-y-privilegios.sql

- a. Conectar al servidor con el usuario del taller12. Login: **andrea**, passwd: **andrea42**
- b. Intentar consultar la tabla clientes.
- c. Salir y conectar con el usuario: postgres
- d. Conectar a la BD dbtaller12.
- e. Asignar permiso select a la tabla de clientes para andrea.
- f. Volver a conectar con andrea y probar de nuevo el select y probar también un delete sobre la misma tabla.

- g. Otorgar todos los permisos sobre la tabla clientes. Probar una actualización de la edad con el usuario andrea.
- h. Conectar a la BD dbtaller12 con el usuario postgres y crear un nuevo ROLE
 - i. Create role pedro LOGIN PASSWORD 'pedro123';
- i. Crear permisos a nivel de columna para el usuario pedro
- j. Restringir el acceso a los teléfonos de la tabla de clientes.
- k. Conectar con pedro a dbtaller12 y probar select con una o varias columnas, incluyendo la col: teléfono.

5) Disponemos del fichero: practicas/base-taller-013-roles-y-privilegios.sql

- a. Dentro de la BD dbTaller13 hay una tabla de clientes
- b. Los siguientes roles:
 - i. daniel / daniel123 que hereda de:
 - ii. consulta / consulta123
- c. Quitar todos los permisos del Role consulta sobre la tabla de clientes
- d. Volver a conectar con este y comprobarlo
- e. Después probar con el usuario **daniel** que pertenecía al **ROLE** consulta. Veremos que no funciona porque el role daniel pertenece al role consulta.
- f. Ahora otorgar los permisos a daniel para que tenga acceso a la tabla clientes
- g. Después conectar con el usuario daniel y otorgarle permisos y ahora si debería, de tener acceso.
- h. Después conectar con consulta y probar, no debería de tener acceso, pero si daniel.
- i. Después volver a dar permisos sobre la tabla de clientes para el role consulta y probarlo.
- j. Después revocar los permisos a daniel.
- k. Y probar el usuario daniel. Veremos que sigue teniendo permisos sobre la tabla de clientes porque daniel heredaba de consulta. Cuando damos permisos al role de consulta automáticamente también los tiene daniel.

6) Conectar con el superusuario, crear un role rafael con login y password: rafael123.

- a. Otorgar el privilegio createdb a este rol.
- b. Crear una nueva base de datos: dbtaller14.
- c. Salir de la sesión actual y conectar con el role: rafael

- d. Conectar a la BD dbtaller14 y crear una tabla productos con un par de cols. Se puede crear con pgAdmin indicando que el dueño es rafael.
- e. Consultar las tablas de la BD dbtaller14:

```
dbtaller14=> \dt
          Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public  | productos | tabla | rafael
(1 fila)
```

- f. Iniciar sesión con el usuario **postgres**
- g. Intentar eliminar el role rafael con drop role.

```
dbtaller14=# drop role rafael;
ERROR:  no se puede eliminar el rol «rafael» porque otros objetos dependen de él
DETALLE:  dueño de secuencia productos_id_seq
dueño de tabla productos
```

- h. Reasignar todos los objetos del role **rafael** al role **postgres**.

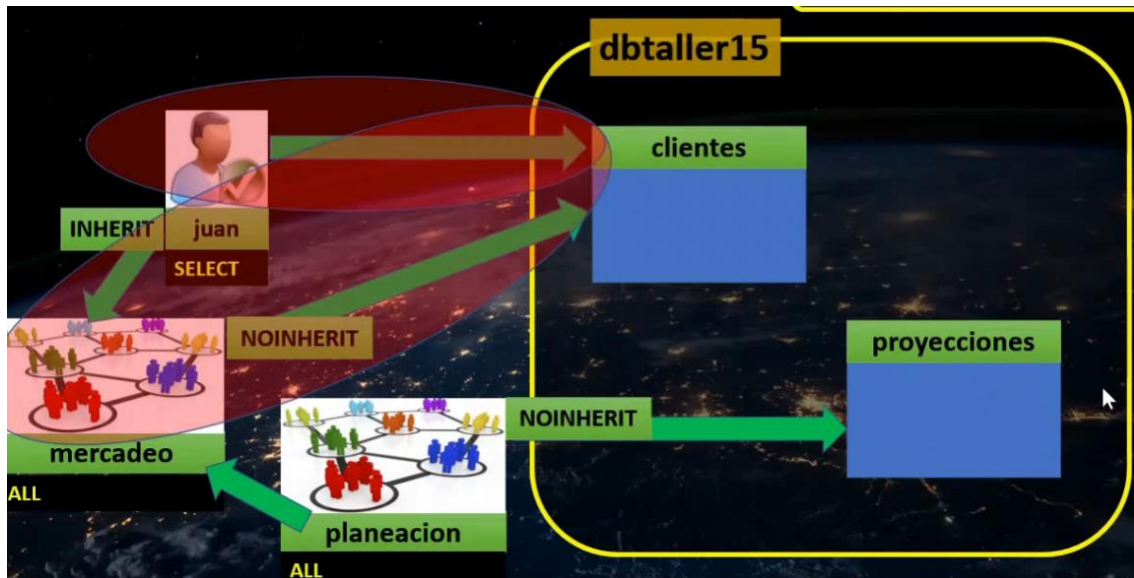
```
dbtaller14=# \dt
          Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public  | productos | tabla | postgres
(1 fila)
```

- i. Eliminar los objetos del propietario **rafael**
- j. Intentar eliminar el role rafael con drop role → debería borrarlo.

7) Roles en grupo

- a. Desde psql crear una nueva BD dbtaller15 con el usuario postgres.
- b. Conectar con la BD dbtaller15.
- c. Crear una tabla: proyecciones (año integer, mes integer, valor integer)
- d. Crear una tabla: clientes (yy integer, mm integer, valor integer)
- e. Crear un role: juan que herede, pueda hacer login y la password 'juan123'.
- f. Añadir permiso select a juan a la tabla de clientes
- g. Comprobar el resultado del comando: \z
- h. Crear un role de grupo con el nombre mercadeo (noinherit)
- i. Otro segundo role de grupo con el nombre planeación (noinherit)
- j. Otorgar al role mercadeo todos los privilegios de la tabla clientes
- k. Otorgar al role planeacion todos los privilegios de la tabla proyecciones
- l. Comprobar lo que muestra el comando: \z
- m. Añadir el role juan al role de grupo mercadeo y también planeacion.

- i. Grant planeación to juan, ...
- n. Añadir el grupo mercadeo al grupo de planeación.
 - i. Grant mercadeo to planeación;
- o. Conectar con el role juan
- p. Set role mercadeo (en una sesión con juan).
- q. Probar a insertar un cliente con el usuario de juan. Y lo hace bien.



- r. Probar a insertar un registro en la tabla de proyecciones. Pero no tiene permiso.
- s. Ejecutar sobre el usuario de juan → SET ROLE planeación;
- t. Y volvemos a probar. Ahora si puede.
- u. Después intentamos select sobre clientes. No puede
- v. Con set role nombre_role sobre el user juan podemos ir cambiando el role y tomará los permisos del role asignado.