**DCL:**

**Per iniciar template1 / training amb el noste usuari:**

| isx48062351@i24:~$ psql template1; psql (13.3 (Debian 13.3-1)) Type "help" **for** **help**.  template1=> |
| --- |

**Intentem crear un usuari amb la sessió del nostre user, però comprovem que no podem:**

| template1=> create user pepito; ERROR: permission denied **to** create role template1=> |
| --- |

**En la sessió de postgres si que podem:**

| $ sudo -**u** postgres psql |
| --- |

**DIFERÈNCIES ENTRE ‘CREATE USER’ I ‘CREATE ROLE’:**

**CREATE USER:** Cas concret de ‘CREATE ROLE’ (‘ALIAS’ per fer-ho amb ‘LOGIN’) → Els ‘users’ son ‘ROLES’ amb ‘LOGIN’ assignat.

**CREATE ROLE:** Cas general.

**Podem cambiar de ‘ROLE’ dins d’una sessió amb el nostre usuari o des de fora.**

**PER BORRAR UNA TAULA / USUARI UTILITZEM ‘DROP’:**

| **drop** table <taula> **drop** user <user> |
| --- |

**PER MODIFICAR UTILITZEM ‘ALTER’:**

| **ALTER USER** <user> WITH **CREATEROLE**; |
| --- |

**Per assignar un usuari a un grup/role, o per fer que un grup / role incorpori els privilegis d’un altre rol utilitzem ‘GRANT’:**

| grant role\_pare [, ...] **to** role\_fill [, ...]; |
| --- |

**A grup fill li atorguem els privilegis definits al grup pare:**

| grant vendes **to** <usuari1>, <usuari2>; |
| --- |

**usuari1 i usuari2 tindran els privilegis definits a vendes.**

**Per treballar amb els privilegis d’un rol en concret utilitzem ‘ROLE’:**

| **set** role <vendes>; |
| --- |

**Per veure com a quin usuari / role estàs treballant:**

| SELECT session\_user, current\_user; |
| --- |

**Per tornar al vostre role normal:**

| reset role; |
| --- |

**Per treballar des d’un rol concret, podeu iniciarla sessió amb psql amb aquell rol, o des dela sessió on esteu, podeu canviar de rol.**

**Per veure la llista de rols existents:**

| \du |
| --- |

**Veiem per exemple que els usuaris tenen LOGIN i els rols no.**

**Gestió de privilegis. GRANT / REVOKE:**

| GRANT privileges ON object TO role|PUBLIC; REVOKE privileges ON object FROM role|PUBLIC; |
| --- |

**Permetre la gestió de totes les dades de totes les taules a l'usuari:**

| GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO <usuari>; |
| --- |

PostgreSql permet diversos tipus d'autenticació definits al fitxer **/etc/postgresql/13/main/pg\_hba.conf** (o **/var/lib/pgsql/data/pg\_hba.conf** en versions antigues) A cada línia d'aquest fitxer s'especifica:

●si s'accedeix localment o remotament

●a quines bases de dades es pot accedir

●quins usuaris hi poden accedir

●quines adreces (IP) poden accedir

●el mètode d'autenticació

Per fer accessos remots, hem de dir a Postgresql que escolti a les adreces de xarxa que volem en el fitxer **/var/lib/pgsql/data/postgresql.conf:**

**# Per defecte:**

listen\_addresses='localhost'

**# Per escoltar tot:**

listen\_addresses='\*'

**# Per escoltar IPs determinades:**

listen\_addresses='192.168.0.24, 192.168.0.26'

**CADA COP que toquem fitxers de configuració hem de reiniciar el servei de Postgresql:**

# systemctl restart postgresql

**Accés:**

$ psql -h <ip\_servidor|nom\_servidor> -U <nom\_usuari> <nom\_bd>

**PER VEURE les BBDD:**

training=> \l

**PER VEURE els USUARIS:**

training=> \du

**PER VEURE els PRIVILEGIS:**

training=> \dp

**peer:** Tothom que es connecti en local utilitzarà la validació de sistema.

**ident:** L’autenticació és basa en el sistema operatiu, si l’usuari existeix, ja haurà demanat l’autenticació per entrar a la xarxa, llavors, serà usuari existent, i li deixarà entrar,

**md5:** Demanarà usuari i password.

**trust:** No demanarà res.

**reject:** No deixarà entrar.

Podem restrinigr (**PER IP**) que només el servidor web consulti les dades de la BBDD.

Podem accedir a la BBDD d’un company/a mitjançant l’accés remot, hem de modificar el següent fitxer: **/etc/postgresql/13/main/postgresql.conf**

Descomentar valor listen\_addresses i posar-li el valor **\***

**listen\_addresses = '\*'**

Descomentar el valor port

**port=5432**

Modificar el fitxer **/var/lib/pgsql/data/pg\_hba.conf** i afegir una línia a l'apartat IPv4 local connections: Per exempl**e:**

**host all all 192.168.0.83/32 trust**

­­(D'aquesta manera només permetem l'accés a la base de dades al pc amb la ip 192.168.0.83. Comprova la direcció sencera. Si en comptes de posar /32, posem /24 només comprovaria els 3 primers grups i deixaria entrar totes les ip de la classe: 192.168.0.\*)

Fer restart del servei postgres

**# systemctl restart postgresql**

**Des del client**

Ara ja podem entrar a la base de dades remotament

**# psql ­-h 192.168.0.83 ­-p 5432 -U postgres training**

**passwd →** Ens permet cambiar la contrasenya DINS DE POSTGRES!

host all guest 10.200.244.229/32 pam **→ NO ENS DEIXARÀ PERQUE GUEST ÉS UN USUARI LOCAL, NO ESTÀ REGISTRAT AL SERVIDOR (GANDHI)**

host all isx46994723 10.200.244.229/32 pam→**ENS DEIXA PERQUE SOM UN USUARI REGISTRAT AL SERVIDOR (GANDHI)**

**---------------------------------------------------------------------------------------------------------------------------**

**CREATE TABLE** vehicles (

id\_vehicle **varchar(10) PRIMARY KEY,**

sale\_date **date,**

available **boolean DEFAULT true**

**<etiqueta> <tipus> <NULL/NOT NULL> <camp per defecte(default) + valor>**

**);**

**local all all 192.168.0.83/32 trust**

**1 -** type (local / host)

**2 -** database

**3 -** user

**4 -** ip-addres + mask

**5 -** method

\df → llistar funcions

\x → expandir SELECT