



### Tarea 3: Ejercicio en PostgreSQL

Instrucciones: Crear un usuario con contraseña que su cuenta sea válida por un mes y establecer un límite de conexiones. Además, crear un role, asignar permisos de lectura, actualización y borrado en una tabla llamada “estudiante” y asignar dicho role al usuario del punto anterior.

Primero creamos el usuario, para ello se utilizó el comando: `CREATE ROLE <usuario> WITH PASSWORD '<contraseña>' VALID UNTIL 'fecha' CONNECTION LIMIT n;` donde:

- `CREATE ROLE <usuario> WITH PASSWORD '<contraseña>'`: hace referencia a la creación de un usuario con contraseña.
- `VALID UNTIL 'fecha'`: Se utiliza para que la cuenta sea válida hasta una fecha fijada por nosotros.
- `CONNECTION LIMIT n`: Se utiliza para dar una n cantidad de conexiones a un usuario

Para esta actividad se creó un usuario llamado pepe, con contraseña, válido hasta el mes siguiente y con un límite de 100 conexiones:

```
postgres=# CREATE ROLE pepe WITH LOGIN PASSWORD 'notapassword' VALID UNTIL '2021-10-07' CONNECTION LIMIT 100;
CREATE ROLE
postgres=# \du
```

Nombre de rol	Lista de roles	Atributos	Miembro de
pepe	100 conexiones		+
	Contraseña válida hasta 2021-10-07 00:00:00-05		{}
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS		{}

Para consultar el usuario creado se usó el comando `\du` y vemos que fue creado como se especificó.

Luego se procedió a hacer una tabla vacía llamada “estudiantes” utilizando el comando `CREATE TABLE ();`

Para poder consultar la tabla creada se utilizó el comando `\dt`:

```
postgres=# CREATE TABLE estudiante();
CREATE TABLE
postgres=# \dt
               Listado de relaciones
 Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
 public | estudiante | tabla | postgres
(1 fila)
```

Una vez creada la tabla se creó un role llamado “ssa” con el comando CREATE ROLE, PostgreSQL nos avisa que se creó el usuario devolviéndonos un “CREATE ROLE”. Para verificar se escribió el comando \du para ver el role creado:

```
postgres=# CREATE ROLE ssa;
CREATE ROLE
postgres=# \du
               Lista de roles
Nombre de rol | Atributos | Miembro de
-----+-----+-----
 pepe         | 100 conexiones | {}
              | Contraseña válida hasta 2021-10-07 00:00:00-05 |
 postgres    | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {}
 ssa         | No puede conectarse | {}
```

una vez creado el role le dimos los privilegios de escritura, actualización y borrado a la tabla “estudiante” antes creada.

Esto se hizo con el siguiente comando: GRANT SELECT, UPDATE, DELETE ON TABLE estudiante TO ssa;

```
postgres=# GRANT SELECT, UPDATE, DELETE ON TABLE estudiante TO ssa;
GRANT
```

Finalmente se asigna el role al usuario creado anteriormente, esto se hace con el comando GRANT:

```
postgres=# GRANT ssa TO pepe;
GRANT ROLE
```

Tabulamos el comando \du para verificar que el role fue asignado:

```
postgres=# \du
               Lista de roles
Nombre de rol | Atributos | Miembro de
-----+-----+-----
 pepe         | 100 conexiones | {ssa}
              | Contraseña válida hasta 2021-10-07 00:00:00-05 |
 postgres    | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {}
 ssa         | No puede conectarse | {}
```

Nótese que en la tercera columna se especifica para el usuario pepe que es usuario del grupo ssa el cual es el role asignado, por lo que se da por concluido el ejercicio.