

Crear un usuario con contraseña, que su cuenta sea válida por un mes y establecer un límite de conexiones.

Como se ha visto en clase de teoría, lo que se debe de realizar primero es crear un usuario y después asignarle una contraseña, ahora lo que cambia es encontrar algún comando para establecer un límite de conexiones.

Los pasos que se deben de realizar son los siguientes:

- Lo primero que se debe de realizar es entrar a SQL Shell y entrar con el usuario que hemos predeterminado:
- Después crearemos una nueva base de datos, en este caso le asigne el nombre *EjemploTarea3*, para realizar lo anterior se utilizó el comando **CREATE DATABASE EjemploTarea3**;
Donde después tarda un momento en crear la nueva bases de datos y la terminal nos lanzará un mensaje que ha sido creada, **CREATE DATABASE**:
- Si queremos ver qué bases de datos tenemos creadas, basta con el comando **\l**, aquí encontraremos la nueva base de datos que creamos en el paso anterior:

```
psql (13.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# CREATE DATABASE EjemploTarea3;
CREATE DATABASE
postgres=# \l

          Listado de base de datos

```

Nombre	Dueño	Codificación	Collate	Ctype	Privilegios
ejemplotarea3	postgres	UTF8	Spanish_Mexico.1252	Spanish_Mexico.1252	
postgres	postgres	UTF8	Spanish_Mexico.1252	Spanish_Mexico.1252	
template0	postgres	UTF8	Spanish_Mexico.1252	Spanish_Mexico.1252	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	Spanish_Mexico.1252	Spanish_Mexico.1252	=c/postgres + postgres=CTc/postgres

```
(4 filas)

postgres=#
```

- A continuación, se ponemos el comando **\c**, seguido del nombre de la base de datos que creamos para conectarnos y nos lanzará el siguiente mensaje:

```
postgres=# \c ejemplotarea3
Ahora está conectado a la base de datos «ejemplotarea3» con el usuario «postgres».
ejemplotarea3=#
```

- Y para concluir creamos un usuario con su contraseña, seguido que su cuenta sea válida por un mes y estableciendo límites de conexiones, en este caso le daré hasta 250 conexiones al usuario: sid.

El comando correspondiente será:

```
CREATE USER sid WITH PASSWORD 'parte1' VALID UNTIL '2021-9-9'
CONNECTION LIMIT 250;
```

```
ejemplotarea3=#
ejemplotarea3=# CREATE USER sid WITH PASSWORD 'parte1' VALID UNTIL '2021-9-9' CONNECTION LIMIT 250;
ejemplotarea3=#
```

- Si queremos que fue creado el rol de forma exitosa, ponemos el comando `\du` y listo:

```
ejemplotarea3=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}
sid	250 conexiones Contraseña válida hasta 2021-09-09 00:00:00-05	+ {}

```
ejemplotarea3=#
```

Crear un role, asignar permisos de lectura, actualización y borrado en una tabla de nombre “estudiante”. Asignar dicho role al usuario del paso anterior.

- Creamos un nuevo role y este será para la parte 2, con el comando: **CREATE ROLE P2;**
- Después creamos la tabla de nombre estudiante: **CREATE TABLE estudiante();**
- Para saber lo que llevamos hasta el momento y saber si fue creada la tabla con éxito basta con el comando `\du` y listo:
- Después para saber el contenido de la tabla estudiante ponemos el comando `\dp`

```
ejemplotarea3=# CREATE TABLE estudiante();
ejemplotarea3=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}
sid	250 conexiones Contraseña válida hasta 2021-09-09 00:00:00-05	+ {}

```
ejemplotarea3=#
```

```

ejemplotarea3=# \dp

```

Esquema	Nombre	Tipo	Privilegios	Privilegios de acceso a columnas	Políticas
public	estudiante	tabla			

(1 fila)

- Después para agregar permisos al rol y rol a usuario:
- Usamos el comando **GRANT SELECT, UPDATE, DELETE ON estudiante to P2;**
Es importante este comando ya que **GRANT** es el que se encarga de otorgar los privilegios a los roles.

SELECT es para asignar el permiso de lectura y **DELETE** para borrar.

```

ejemplotarea3=# CREATE ROLE P2;
CREATE ROLE
ejemplotarea3=#
ejemplotarea3=#
ejemplotarea3=# GRANT SELECT, UPDATE, DELETE ON estudiante to P2;
GRANT
ejemplotarea3=#
ejemplotarea3=#
ejemplotarea3=# \dp

```

Esquema	Nombre	Tipo	Privilegios	Privilegios de acceso a columnas	Políticas
public	estudiante	tabla	postgres=arwdDxt/postgres+ p2=rwd/postgres		

(1 fila)

```

ejemplotarea3=# \du

```

Nombre de rol	Lista de roles Atributos	Miembro de
p2	No puede conectarse	{}
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}
sid	250 conexiones	{}
	Contraseña válida hasta 2021-09-09 00:00:00-05	

```

ejemplotarea3=#

```