

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

Crear un rol y asignar permisos de lectura, actualización y borrado en una tabla de nombre "estudiante", Asignar dicho rol al usuario del paso anterior

Crear usuario con contraseña, cuenta valida por un mes y establecer un limite de conexiones:

```
postgres=# CREATE USER Tarea4 WITH ENCRYPTED PASSWORD '123' VALID UNTIL '2022-03-25' CONNECTION LIMIT 5;
CREATE ROLE
postgres=#
```

Tabla de usuarios:

```
postgres=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{ }
tarea4	5 conexiones	+ { }
	Contraseña válida hasta 2022-03-25 00:00:00-06	

Creación de tabla "estudiante":

```
postgres=# CREATE TABLE estudiante(Atributo varchar(50));
CREATE TABLE
postgres=# \d
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public | estudiante | tabla | postgres
(1 fila)
```

Creación de rol "Ejemplo":

```
postgres=# CREATE ROLE Ejemplo;
CREATE ROLE
postgres=#
```

Asignar permisos al Rol "Ejemplo":

```
postgres=# GRANT SELECT ON estudiante TO Ejemplo;
GRANT
postgres=# GRANT UPDATE ON estudiante TO Ejemplo;
GRANT
postgres=# GRANT DELETE ON estudiante TO Ejemplo;
GRANT
postgres=#
```

Por ultimo se asigna el rol “ejemplo”, con sus respectivos privilegios o permisos, al usuario creado inicialmente, verificamos en el listado:

```
postgres=# GRANT Ejemplo to Tarea4;  
GRANT ROLE  
postgres=#
```

```
postgres=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
ejemplo	No puede conectarse	{}
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}
tarea4	5 conexiones	+ {ejemplo}
	Contraseña válida hasta 2022-03-25 00:00:00-06	

```
postgres=#
```