



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
FACULTAD DE INGENIERÍA  
INGENIERÍA EN COMPUTACIÓN



BASES DE DATOS  
GRUPO: 01

TAREA #3

Alumno:  
López González Kevin

Profesor:  
ING. Fernando Arreola Franco

9 de septiembre de 2021

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

La instrucción completa es:

```
CREATE USER kevin WITH PASSWORD '123abc' VALID UNTIL '2021-10-09' CONNECTION LIMIT 5;
```

Donde:

- CREATE USER kevin: Indica que queremos crear un nuevo usuario de nombre 'kevin'.
- WITH PASSWORD '123abc': La asignamos la contraseña '123abc'.
- VALID UNTIL '2021-10-09': Indicamos que la contraseña será válida hasta el 09 de octubre del 2021.
- CONNECTION LIMIT 5: Limita a 5 el número de conexiones.

```
grupo1=#
grupo1=# CREATE USER kevin WITH PASSWORD '123abc' VALID UNTIL '2021-10-09' CONNECTION LIMIT 5;
CREATE ROLE
grupo1=#
grupo1=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
kevin	5 conexiones Contraseña válida hasta 2021-10-09 00:00:00-05	+ {}
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}

Figura 1: Creación del usuario y verificación de operación.

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

Primero, creamos el role llamado 'admin' con la siguiente instrucción:

```
CREATE ROLE admin;
```

```
grupo1=#
grupo1=# CREATE ROLE admin;
CREATE ROLE
grupo1=#
grupo1=# \du
```

Nombre de rol	Lista de roles Atributos	Miembro de
admin	No puede conectarse	+ {}
kevin	5 conexiones Contraseña válida hasta 2021-10-09 00:00:00-05	+ {}
postgres	Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}

Figura 2: Creación del role y verificación de operación.

Es necesario tener una tabla para poder asignar permisos sobre ella, por lo tanto la crearemos con la siguiente instrucción bajo el nombre 'estudiante':

```
CREATE TABLE estudiante(nombre varchar(50));
```

```

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

```

Figura 3: Creación de la tabla y verificación de operación.

Con la tabla creada, asignamos los permisos al role con la siguiente instrucción:

```
GRANT SELECT, UPDATE, DELETE ON estudiante TO admin;
```

```

grupol=#
grupol=# GRANT SELECT, UPDATE, DELETE ON estudiante TO admin;
GRANT
grupol=#

```

Figura 4: Asignación de permisos al role.

Antes de asignar el role, comprobamos que el usuario 'kevin' no tiene permisos sobre la tabla 'estudiante'.

```

Conectándose para el usuario «kevin»
Ahora está conectado a la base de datos «grupol» con el usuario «kevin».
grupol=> \du
                Lista de roles
Nombre de rol | Atributos | Miembro de
-----+-----+-----
admin         | No puede conectarse | {}
kevin         | 5 conexiones | {}
postgres      | Constraseña válida hasta 2021-10-09 00:00:00-05 | {}
              | Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS | {}

grupol=> SELECT * from estudiante;
ERROR: permiso denegado a la tabla estudiante
grupol=> UPDATE estudiante SET nombre = 'Bruno' WHERE nombre = 'Armando';
ERROR: permiso denegado a la tabla estudiante
grupol=> DELETE FROM estudiante WHERE nombre = 'Isaac';
ERROR: permiso denegado a la tabla estudiante
grupol=>

```

Figura 5: Verificación de la ausencia de permisos.

Asignamos el role 'admin' al usuario 'kevin' con la instrucción:

```
GRANT admin TO kevin;
```

```

grupo1=# GRANT admin TO kevin;
GRANT ROLE
grupo1=#
grupo1=# \du

```

Nombre de rol	Lista de roles Atributos	Miembro de
admin	No puede conectarse	{}
kevin	5 conexiones	+ {admin}
postgres	Contraseña válida hasta 2021-10-09 00:00:00-05 Superusuario, Crear rol, Crear BD, Replicación, Ignora RLS	{}

Figura 6: Asignación del role al usuario y verificación de operación.

Finalmente, observamos que el usuario ya tiene permisos del role sobre la tabla 'estudiante'.

```

grupo1=>
grupo1=>
grupo1=> SELECT * from estudiante;
 nombre
-----
 Ricardo
 Armando
 Isaac
(3 filas)

grupo1=> UPDATE estudiante SET nombre = 'Bruno' WHERE nombre = 'Armando';
UPDATE 1
grupo1=> SELECT * from estudiante;
 nombre
-----
 Ricardo
 Isaac
 Bruno
(3 filas)

grupo1=> DELETE FROM estudiante WHERE nombre = 'Isaac';
DELETE 1
grupo1=> SELECT * from estudiante;
 nombre
-----
 Ricardo
 Bruno
(2 filas)

grupo1=> INSERT INTO estudiante(nombre) values('Eduardo');
ERROR: permiso denegado a la tabla estudiante
grupo1=>

```

Figura 7: Verificación de posesión de permisos.

### 3. Bibliografía

- Amazon Redshift Database Developer Guide [Online]. Available: [https://docs.aws.amazon.com/redshift/latest/dg/redshift-dg.pdf#r\\_CREATE\\_USER](https://docs.aws.amazon.com/redshift/latest/dg/redshift-dg.pdf#r_CREATE_USER)
- UPDATE [Online]. Available: <https://www.postgresql.org/docs/9.1/sql-update.html>
- DELETE [Online]. Available: <https://www.postgresql.org/docs/10/sql-delete.html>