**Aaron Andal : DCL → Data Control Language // sudo -u postgres**
Permite al **Administrador del Gestor de Base de Datos,** el acceso a los ojbetos. Podemos: **otorgar o denegar permisos** a **uno** o **más roles** para hacer determinadas **tareas. Los comandos son GRANT - REVOKE.**

**CREACIÓN USUARIOS**
**postgres@keshi:~$ createuser –interactive (Desde fuera)** → Permite crear de forma interactiva un usuario, desde **fuera**
**postgres@keshi:~$ sudo -u postgres createuser –interactive**
**postgres@keshi:~$ sudo -u postgres createdb keshi** → Crea la BD desde fuera.

**\du** → Lista roles existentes

**CREATE ROLE** nom_user **LOGIN**; → Sin password // **CREATE ROLE** nom_rol **WITH LOGIN PASSWORD** 'password'; → C on password (*No tiene LOGIN por defecto*)
**CREATE USER** nom_user **PASSWORD** 'password'; (**Tiene LOGIN por defecto**)

```
CREATE USER user_name
  [ WITH PASSWORD 'password_value'
  | VALID UNTIL 'expiration' ];

Per exemple:
CREATE USER user_name PASSWORD 'tres45';
És el mateix que:
CREATE ROLE user_name LOGIN PASSWORD 'tres45';
```

```
CREATE ROLE pepito
[ WITH PASSWORD 'password_value'
  | VALID UNTIL 'expiration' ]
  [WITH OPTION SUPERUSER | CREATEDB | CREATEROLE | LOGIN....
];

Per exemple:
CREATE ROLE comercials;
```

**OPCIONES: SUPERUSER / NOSUPERUSER - CREATEDB / NOCREATEDB - CREATEROLE / NOCREATEROLE - INHERIT / NOINHERIT - LOGIN / NOLOGIN - PASSWORD - CONNECTION LIMIT num - VALID UNTIL 'timestamp' - IN ROLE role_name** (*Tendrá sus privilegios*)

**Borrar un usuario (DROP) o modificar (ALTER): DROP USER user. ALTER USER user WITH CREATEROLE;**
**Para ver a que usuario y role estás trabajando → SELECT session_user current_user;**
**Volver al role normal** → RESET ROLE role;
**Cambiar de role en concreto → SET ROLE role;**
**ACCESO:** $ psql -h ip_servidor | nom_servidor -U nom_usuari nom_bd

**PERMISOS (Grant / Revoke)**

## Gestió de privilegis. GRANT / REVOKE

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

On els **privilegis** poden ser:

- SELECT
- INSERT
- UPDATE
- DELETE
- INDEX
- CREATE
- ALTER
- DROP
- GRANT OPTION
- ALL

**GRANT→ AÑADIR PRIVILEGIOS**
**\dp → Ver permisos**

**PERMISOS A TABLAS**

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { [ GROUP ] role_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Ejemplo:
**GRANT SELECT**, **INSERT, DELETE ON** repventas **TO** rrhh; → Se le concede DML i Select.

**GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA** public **TO** tworker; → Se concede todos los privilegios de todas las tablas en public. a tworker.

**PERMISOS A COLUMNAS**

```
GRANT { { SELECT | INSERT | UPDATE | REFERENCES } ( column_name [, ...] )
    [, ...] | ALL [ PRIVILEGES ] ( column_name [, ...] ) }
    ON [ TABLE ] table_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Ejemplo:
**GRANT SELECT, UPDATE** (objectiu, vendes) **ON** oficinas **TO** director;

**PERMISOS A BASES DE DATOS**

```
GRANT { { CREATE | CONNECT | TEMPORARY | TEMP } [, ...] | ALL [ PRIVILEGES ] }
    ON DATABASE database_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

**PERMISOS A ESQUEMAS**

```
GRANT { { CREATE | USAGE } [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

**PERMISOS DE UN ROL A OTRO (INHERIT)**

```
GRANT rol_pare [, ...] TO rol_fill [, ...]
```

**Ejemplo (Permisos de una BD al usuario tmanager y al usuario tadmin): OWNER**

```
CREATE ROLE tgroup;
ALTER DATABASE training OWNER TO tmanager;
ALTER TABLE clientes OWNER TO tmanager;
ALTER TABLE oficinas OWNER TO tmanager;
ALTER TABLE repventas OWNER TO tmanager;
ALTER TABLE productos OWNER TO tmanager;
ALTER TABLE pedidos OWNER TO tmanager;
GRANT tgroup TO tmanager, tadmin;
```

**REVOKE → QUITAR PRIVILEGIOS**

**QUITAR PERMISOS A TABLAS**

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
        | ALL TABLES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

**QUITAR PERMISOS A COLUMNAS**

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | REFERENCES } ( column_name [, ...] )
    [, ...] | ALL [ PRIVILEGES ] ( column_name [, ...] ) }
    ON [ TABLE ] table_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

**QUITAR PERMISOS A BASES DE DATOS**

```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE | CONNECT | TEMPORARY | TEMP } [, ...] | ALL [ PRIVILEGES ] }
    ON DATABASE database_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

**QUITAR PERMISOS DE ESQUEMAS**

```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE | USAGE } [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

**QUITAR PERMISOS DE UN ROL A OTRO**

```
REVOKE [ ADMIN OPTION FOR ]
    role_name [, ...] FROM role_name [, ...]
    [ CASCADE | RESTRICT ]
```

**CAMBIAR PROPIETARIO DE UNA DB: ALTER DATABASE db_name OWNER TO username;**
**CAMBIAR PROPIETARIO DE UNA TABLA: ALTER TABLE table_name OWNER TO username;**

**\dp → Ver privilegios o \dp tabla**

```
rolename=xxxx -- privileges granted to a role
        =xxxx -- privileges granted to PUBLIC

            r -- SELECT ("read")          U -- USAGE
            w -- UPDATE ("write")         C -- CREATE
            a -- INSERT ("append")        c -- CONNECT
            d -- DELETE                   T -- TEMPORARY
            D -- TRUNCATE              arwdDxt -- ALL PRIVILEGES (for tables, varies for other objects)
            x -- REFERENCES
            t -- TRIGGER                /yyyy -- role that granted this privilege
            X -- EXECUTE
```

**SCHEMAS → Permiten agrupar tablas dentro de una base de datos y tener más seguridad y consistencia.**
**\dn → Ver los esquemas**
**\? → Ayuda postgres**

**SCHEMA PUBLIC** → Subcontenedor - **Agrupa tablas** dentro de una **BD. Se pueden crear más esquemas.**

**SELECT session_user, current_user; → PARA VER EL USUARIO ACTUAL.**
**CREATE SCHEMA keshi;**

**SHOW SEARCH_PATH: → Muestra el PATH a buscar.**

| | | |
|---|---|---|
| keshi=# CREATE SCHEMA keshi;<br>CREATE SCHEMA<br>keshi=# SHOW search_path;<br>search_path<br>----------<br>"$user", public<br>(1 row) | keshi=# \dn<br>List of schemas<br>Name \| Owner<br>--------+----------<br>keshi \| keshi<br>public \| postgres<br>(2 rows) | SELECT public.rep_vendes.nom<br>FROM public.rep_vendes; |