

# Comandos más usados en PostgreSQL

## *Comandos de navegación y consulta de información*

- `\c` Saltar entre bases de datos
- `\l` Listar base de datos disponibles
- `\dt` Listar las tablas de la base de datos
- `\d <nombre_tabla>` Describir una tabla
- `\dn` Listar los esquemas de la base de datos actual
- `\df` Listar las funciones disponibles de la base de datos actual
- `\dv` Listar las vistas de la base de datos actual
- `\du` Listar los usuarios y sus roles de la base de datos actual

## *Comandos de inspección y ejecución*

- `\g` Volver a ejecutar el comando ejecutando justo antes
- `\s` Ver el historial de comandos ejecutados
- `\s <nombre_archivo>` Si se quiere guardar la lista de comandos ejecutados en un archivo de texto plano
- `\i <nombre_archivo>` Ejecutar los comandos desde un archivo
- `\e` Permite abrir un editor de texto plano, escribir comandos y ejecutar en lote. `\e` abre el editor de texto, escribir allí todos los comandos, luego guardar los cambios y cerrar, al cerrar se ejecutarán todos los comandos guardados.
- `\ef` Equivalente al comando anterior pero permite editar también funciones en PostgreSQL

## *Comandos para debug y optimización*

- `\timing` Activar / Desactivar el contador de tiempo por consulta

## Comandos para cerrar la consola

-  Cerrar la consola



es importante entender que existe una costumbre no oficial al momento de escribir consultas; consiste en poner en mayúsculas todas las palabras propias del lenguaje SQL como **CREATE, SELECT, ALTER**, etc y el resto de palabras como los nombres de las tablas, columnas, nombres de usuarios, etc en minúscula.

## Ejemplo de creación de una base de datos

```
postgres=# CREATE DATABASE transporte;
CREATE DATABASE
postgres=# \c transporte
Ahora está conectado a la base de datos «transporte» con el usuario «postgres».
transporte=# CREATE TABLE tren ( id serial NOT NULL, modelo character varying, capacidad integer, CONSTRAINT tren_pkey PRIMARY KEY (id) );
CREATE TABLE
transporte=# \d tren
```

Columna	Tipo	Ordenamiento	Nullable	Por omisión
id	integer		not null	nextval('tren_id_seq'::regclass)
modelo	character varying			
capacidad	integer			

```
ndices:
    "tren_pkey" PRIMARY KEY, btree (id)

transporte=# \d tren_id_seq
```

Tipo	Inicio	Mínimo	Máximo	Incremento	¿Cicla?	Cache
integer	1	1	2147483647		1	no

```
Asociada a: public.tren.id
```

```
transporte=# INSERT INTO tren( modelo, capacidad ) VALUES ('Volvo 1', 100);
INSERT 0 1
transporte=# SELECT * FROM tren;
 id | modelo | capacidad 
----+-----+-----
  1 | Volvo 1 |       100
(1 fila)

transporte=#
```

```

transporte=# UPDATE tren SET modelo = 'Honda 0726' Where id = 1;
UPDATE 1
transporte=# SELECT * FROM tren;
 id | modelo | capacidad
-----+-----+-----
  1 | Honda 0726 |      100
(1 fila)

transporte=#

```

```

transporte=# DELETE FROM tren WHERE id = 1;
DELETE 1
transporte=# SELECT * FROM tren;
 id | modelo | capacidad
-----+-----+-----
(0 filas)

transporte=#

```

```

transporte=# \timing
El despliegue de duración está activado.
transporte=# SELECT MD5('Vamos a encriptar este texto');
          md5
-----
443fd0b8b498e0a2b864dd99a3e9384a
(1 fila)

Duración: 0,375 ms
transporte=#

```