

---

**M2.UF4**

**INSTALACION EN LINUX Y**

**EJECUCIÓN DE POSTGRESQL**

**EN MODO TERMINAL**

**Eduard Lara**

# INDICE

---

1. Introducción
2. Tablas
3. Tablas
4. Métodos
5. Nested tables

# INSTALACION EN LINUX

- apt-get install postgresql
- Se requieren permisos de sudoer
- Al instalarse se crea el usuario postgres. Es el usuario que se crea por defecto. Tiene todos los permisos y carece de contraseña.
- Archivos:
  - De configuración: /etc/postgresql/..
  - Binarios: /usr/lib/postgresql
  - De datos: /var/lib/postgresql
  - De registro: /var/log/postgresql

# ACCESO A POSTGRESQL

- La única manera de acceder al sistema es utilizando el usuario de sistema (atención) postgres
- Hacemos ‘su postgres’
- Después llamamos al cliente psql: ‘psql’
- De momento utilizaremos esta forma de acceder a postgresql. Más adelante veremos otras opciones.

# UTILIZANDO PSQL

- psql es un cliente de consola de la base de datos postgresql
- Multitud de funcionalidades. Mucho más completo que el cliente de 'mysql'
- Para opciones de conexión podemos verlas con 'psql --help'. Las opciones son muy similares a las de 'mysql'. En principio, si no indicamos nada, psql se conectará al servidor que rueda en la máquina localhost.
- Una vez iniciado el cliente podemos teclear 'help' para obtener ayuda.
- Cualquier sentencia o instrucción acaba con ';' o '\g'
- Para salir de psql: '\q'. La contrabarra identifica las órdenes internas de psql.
- Respecto al prompt:
  - '#' significa que el sistema está listo. Esperando instrucciones

# UTILIZANDO PSQL

- ‘-#’: la sentencia todavía no se ha acabado
- “#”: faltan comillas para acabar la sentencia
- ‘(#’: falta un paréntesis por cerrar.
- psql permite crear variables:
  - Utilizar la orden interna ‘\set’ para crearla.
  - Utilizar el operador ‘:’ para invocarlas
  - De hecho, psql dispone de un montón de variables ya definidas: ‘:DBNAME’, ‘:HOST’, ‘:USER’, etc.
- psql permite procesamiento por lotes: ‘psql postgres -f archivo.psql’
- psql permite indicar el dispositivo y el formato de salida de los datos: html, latex, terminal, etc.

```
postgres=# \set var prueba
postgres=# \echo var
var
postgres=# \echo :var
prueba
```

# GESTIÓN DE LAS BBDD

- Con '\l' psql nos informa de las bbdd disponibles (equivale al 'show databases' de mysql)
- Con '\c' para "conectarnos" a una base de datos (equivale al 'use' de mysql)
- '\d' para listar tablas de una base de datos o para describir los campos de cada tabla. (describe)
- Más opciones:
  - '\dt' → muestra tablas
  - '\di' → muestra lista de índices
  - '\dv' → muestra lista de vistas
  - '\dp \z' → muestra privilegios sobre las tablas

# TIPOS DE DATOS

- Lógicos: boolean o bool; puede asumir el valor true / false.
- Numéricos: smallint, int, bigint, numeric, real, double precision
- Hay un valor numérico especial para implementar series autoincrementales  
→ serial. Ya lo veremos
- Caracteres: char(n), varchar(n), text
- Fechas y horas: time, date, timestamp
- Matrices: formato especial de postgres. Permite albergar más de un valor en una celda de la tabla. Ya volveremos.

# OPERADORES

- AND, OR, NOT
- <, <=, >, >=,<>, !=
- +,-,\* , /,%,...
- || → Concatenar caracteres
- ~ → Correspondencia con expresión regular
- ...

## Problema 1

Crea una tabla con nombre ‘PrimeraTabla’ en la base de datos ‘postgres’ con los siguientes campos:

- Id\_PrimeraTabla: se trata de un número que oscila entre el 1 y el 1000. Será clave de la tabla.
- Nombre: conjunto de caracteres entre 5 y 40
- Descripción: texto
- Fecha inscripción

Después de crear la tabla, ejecuta ‘\d PrimeraTabla’ para ver cómo psql describe la tabla. Inserta varios valores en la tabla y realiza alguna consulta.

## Problema 2

- Averigua cuál es la versión de tu postgres y, a continuación, localiza la documentación oficial de dicha versión.
- Prueba en psql: '\h' y '\?' ¿Qué dos informaciones nos aporta?

## Problema 3

Crea un tabla, que llamarás SegundaTabla, que tenga los siguientes campos:

- Un campo serial (autoincremental) y será clave. Lo llamaremos Id\_SegundaTabla.
- Un número que referenciará al Identificador de la primera tabla: Id\_PrimeraTabla.
- Un campo alfanumérico que llamaremos “dirección postal” con una longitud máxima de 100 caracteres.

Después de crear la tabla, ejecuta ‘\d SegundaTabla’ para ver cómo psql describe la tabla. Inserta varios valores en la tabla y realiza alguna consulta.

## Problema 4

- Muestra el número total de filas de primera tabla
- Muestra las fechas de primera tabla con el formato: 05 de SEP de 2001.  
Utiliza, de momento, nombres de mes en inglés. Ayuda: debes utilizar la función “to\_char”
- Muestra de la dirección de segundatabla los 3 primeros caracteres en mayúsculas
- Realiza un inner join entre las dos tablas.
- Realiza un outer join entre las tablas.
- Muestra aquellas filas de la primeratabla cuyo ‘Nombre’ empiece por ‘M’.