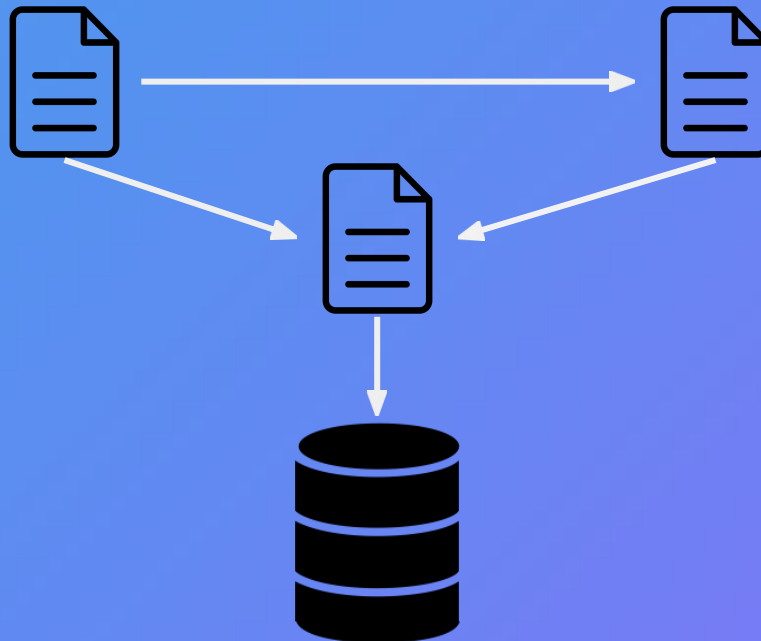




# Base de Datos

# Base de Datos - Introducción

Una Base de Datos es un conjunto de datos organizados y relacionados entre sí, almacenados sistemáticamente, que nos permiten hacer uso de ellos a lo largo del tiempo.



# Base de Datos - Introducción

## Características:

- Datos interrelacionados
- Redundancia mínima
- Acceso concurrente de múltiples usuarios

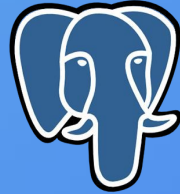
Podemos guardar nuestros datos en otros lugares diferentes a una Base de Datos, como por ejemplo en archivos, pero:

- No es recomendable en una aplicación que maneje muchos datos.
- No es fácilmente mantenible.
- No es performante => Ejemplo: Si quiero solo los clientes con determinadas características, seguramente tenga que leer todo el archivo de clientes, y filtrar en memoria.
- El acceso concurrente se dificulta o se pierde.

## Sistema de Gestión de Base de Datos

- Es un software específico que funciona como interfaz entre el Usuario, la Base de Datos, y las aplicaciones que la utilizan.
- También proporciona las herramientas para poder:
- Definir la estructura de los datos
- Manipular los datos (Crear, consultar, modificar, eliminar)
- Mantener la integridad de la información
- Proporcionar seguridad y privacidad sobre los datos





PostgreSQL

**Postgres es el motor de Base de Datos con el que vamos a trabajar.**

- Es muy robusto
- Tiene muy buena performance
- Es Open Source!
- Es una base de datos relacional

**SQL es el lenguaje (Structured Query Language) que utilizaremos para interactuar con la Base de Datos.**

- Este lenguaje es compartido por la mayoría de las Bases de Datos.
- Es básicamente un lenguaje de programación, y tiene comandos como:
- `SELECT firstName from person`

## Conceptos de una Base de Datos Relacional

- **Entidades:** Objetos acerca de los cuales se guarda información relevante. Ej: Producto
- **Atributos:** Características de la entidad. Ej: Nombre, Precio, Descripción, Estado
- **Tablas:** Objetos de la Base de Datos donde se almacenan los datos. Las tablas tienen columnas y filas. Normalmente una tabla representa una entidad, pero también puede representar la relación entre entidades.
- **Relación:** Es la conexión que puede existir entre dos o más entidades.



## Tipos de comandos SQL

Vamos a distinguir dos tipos de comandos en SQL: **DML** Y **DDL**

- **DML:** (Data manipulation language) Sentencias de manipulación de datos.
- **DDL:** (Data definition language) Sentencias de definición de datos.

### DML

- **SELECT:** Recuperar datos
- **INSERT:** Agregar filas de datos
- **DELETE:** Eliminar filas de datos
- **UPDATE:** Modificar datos

### DDL

- **CREATE:** Crear objetos (Tablas, índices, etc)
- **DROP:** Eliminar objetos
- **ALTER:** Modificar objetos



# Base de Datos

Para crear una tabla por ejemplo, el comando es el siguiente:

```
CREATE TABLE table_name (  
  column1 datatype,  
  column2 datatype,  
  column3 datatype,  
  ....  
);
```

```
CREATE TABLE clients (  
  id integer,  
  first_name text,  
  last_name text,  
  birth_date date  
);
```

El tipo de los datos varía un poco de una Base de Datos a otra, pero básicamente son los mismos, con distintos nombres, y detalles diferentes (*int*, *integer*, *number*).

[Acá](#) pueden encontrar una referencia de los tipos de datos de Postgres.

## Base de Datos

Si queremos ahora agregar el campo ciudad ('city'):

```
ALTER TABLE clients ADD city text;
```

Podríamos también borrar la tabla con:

```
DROP TABLE clients;
```

## Base de Datos

Vamos a agregar datos a nuestra Base de Datos, con el comando INSERT

```
INSERT INTO clients(id, first_name, last_name, birth_date, city) VALUES  
(1, 'Diego', 'Caceres', '1990-12-27', 'Montevideo'),  
(2, 'Pablo', 'Lopez', '1990-05-20', 'Colonia');
```

Podemos comprobar que esto fue insertado con éxito, realizando una query de SELECT (el \* acá funciona como un comodín para traer todas las columnas de la tabla):

```
SELECT * from clients;
```

## Base de Datos

Como podemos hacer para realizar consultas un poco más complejas, por ejemplo:

1. Obtener todos los clientes de una ciudad específica.
2. Obtener el cliente con ID 2.
3. Obtener solo las columnas 'id' y 'first\_name' de los clientes con ID 2 y 3.

Para esto, vamos a utilizar **SELECT** combinado con **WHERE** para especificar la condición que le queremos dar a la Query

```
Select *  
From clients  
Where city = 'Montevideo';
```

```
Select *  
From clients  
Where id = 2;
```

```
Select id, first_name  
From clients  
Where id = 2 OR id = 3
```

## Base de Datos

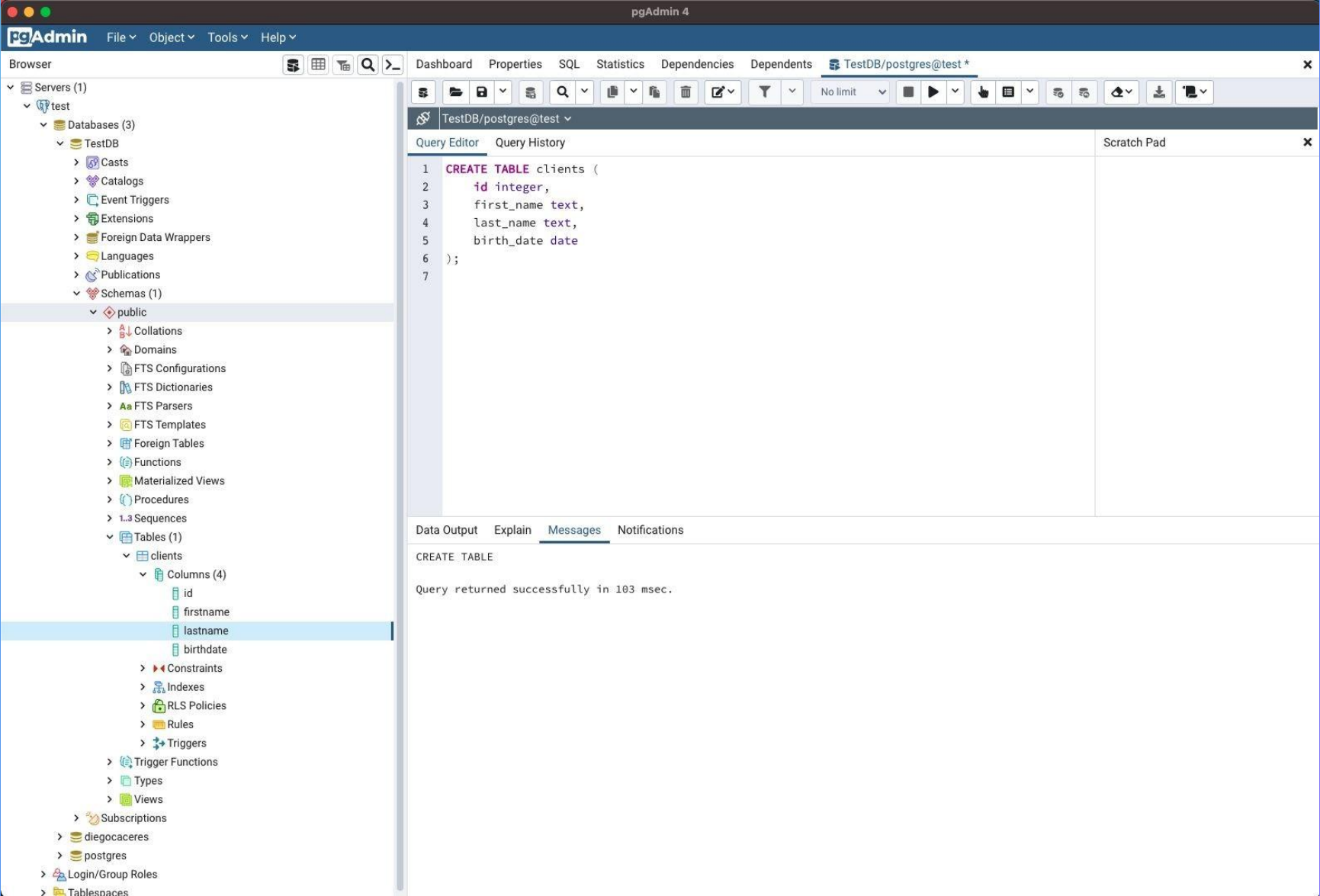
Para la Base de Datos vamos a instalar 2 aplicaciones en Mac. La primera es el motor de Base de Datos en sí, o “servidor de base de datos”:

- MacOS: Descargar e instalar [PostgresApp](#).
- Windows: Descargar e [instalar](#).

Durante la instalación pueden dejar todo por defecto, pero recuerden la contraseña que ingresen cuando les pida, ya que será la contraseña que utilicen para conectarse al motor de base de datos.

Los que instalen en MacOS también van a instalar un programa para manejar la Base de Datos de forma visual llamado “[PGAdmin](#)” (en Windows se instala con el mismo instalador que la Base de Datos).

# Base de Datos



- Instalen en sus computadoras PostgreSQL
- Creen una Base de Datos de prueba, o que luego utilizarán en el proyecto.
- Definan su esquema de Base de Datos, es decir, que tablas van a tener, que atributos va a tener cada tabla.
- Piensen qué tipos de consultas van a necesitar realizar sobre estos datos.

[dbdiagram](#) es una muy útil herramienta para diseñar el esquema de la base de datos.



**GUSTAVO RODRIGUEZ**

FULL STACK DEVELOPER  
SOLCRE



[gustavgueez](#)



[gustavgueez](#)