

# 🎓 Clase 1: Introducción a PostgreSQL y Primeras Consultas

## 👋 Bienvenidos al primer módulo

En esta sesión, exploraremos los conceptos fundamentales de **SQL** y nos sumergiremos en el entorno de **PostgreSQL**. 🚀

## 🧠 ¿Qué aprenderás?

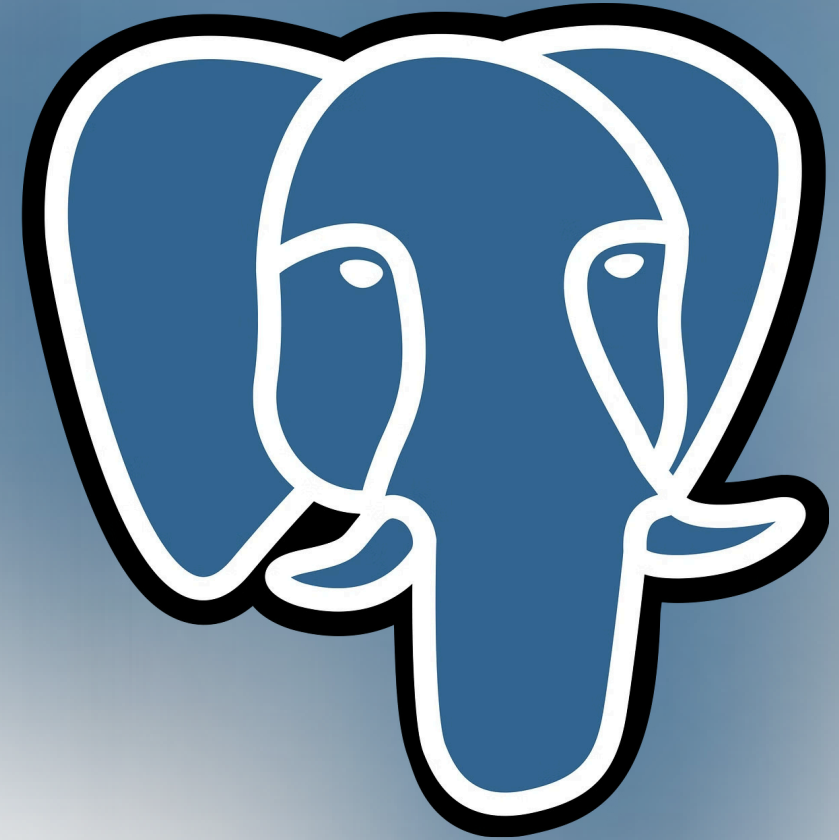
- 📖 Conceptos clave sobre bases de datos.
- 💻 Cómo interactuar con bases de datos en PostgreSQL.
- 🔍 **Ejecutar consultas básicas** para obtener información.

## 🔗 Conexión de PostgreSQL con Python:

- 🛠️ Uso de **psycopg2** para conectar **PostgreSQL** desde Python.

👨‍🏫 **Profesor:** Yoseph Ayala Valencia

☀️ ¡Prepárate para dominar los fundamentos de SQL y llevarlos al siguiente nivel con Python! 🦆



# ¿Qué es SQL? 🤔

## Lenguaje de programación estándar

SQL (Structured Query Language) es un lenguaje de programación estándar utilizado para gestionar y manipular bases de datos relacionales. 🗄️

## Operaciones en bases de datos

Permite realizar operaciones como consultar 🔍, actualizar ✎️, insertar + y eliminar ✖️ datos en sistemas de gestión de bases de datos (DBMS) como MySQL, PostgreSQL, SQL Server y OracleDB. 🛠️

## Esencial para diversos roles

Es esencial para desarrolladores 💻, analistas de datos 📊, administradores de bases de datos 🗄️ y científicos de datos 🧪.




```
53 USE DatabaseName;
54 GO
55
56 CREATE PROCEDURE ProcedureName
57     @FirstName type,
58     @LastName type....
59 AS
60     ....
61     //Your SQL query here
62     Select FirstName, LastName
63     FROM Employee
64     WHERE FirstName = @FirstName AND LastName = @LastName
65 GO
```

# ¿Por qué aprender SQL?

1

## Lenguaje Versátil



SQL no se limita solo a tareas de programación.

Profesionales en marketing , ventas  y finanzas 

pueden usar SQL para analizar datos rápidamente .


2

## Manejo de Grandes Volúmenes de Datos

Con la creciente digitalización , SQL facilita la gestión y el análisis de grandes bases de datos .



3

## Impulso en Ciencia de Datos

SQL es la habilidad inicial y fundamental para quienes buscan una carrera en ciencia de datos .

4




## Habilidad Demandada

SQL es una de las habilidades más solicitadas en el mercado laboral , relevante para diversos roles profesionales .





# ¿Qué es una base de datos (BD)?

## Definición:

Una base de datos es una **colección organizada de datos** que se puede:

-  **Acceder:** Consultar la información almacenada.
-  **Gestionar:** Modificar, eliminar o añadir datos.
-  **Actualizar:** Mantener la información al día de manera eficiente.

## ¿Para qué sirve?

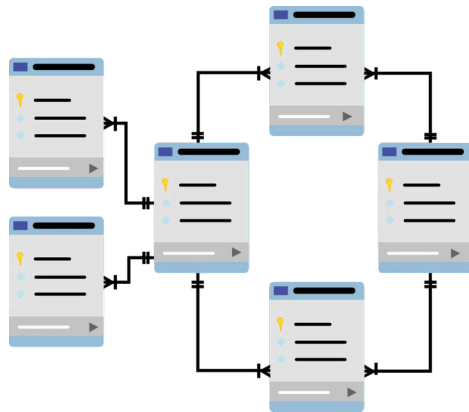
-  **Almacenar** datos de forma estructurada y centralizada.
-  **Recuperar** información fácilmente para su uso.
-  Facilitar **análisis** y toma de decisiones.
-  **Transformar** datos dispersos en **información valiosa**.



# 🔧 Bases de datos relacionales vs no relacionales

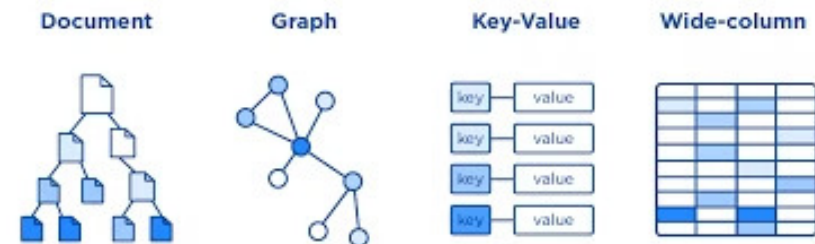
## 📊 Relacionales

- 📋 Datos organizados en **tablas** con filas y columnas.
- 🔗 **Relaciones** entre tablas (clave primaria y clave foránea).
- ✅ Ideales para aplicaciones con estructura de datos **predefinida** y **transacciones**.



## 📁 No relacionales

- 📄 Datos almacenados en estructuras **flexibles**, como documentos, pares clave-valor 🔑, o grafos 🕸.
- ⚡ Mejor **rendimiento** y **escalabilidad** para volúmenes masivos de datos **no estructurados**.



# PostgreSQL: Potencia y Flexibilidad para el Científico de Datos ⚡

## Rendimiento Óptimo

PostgreSQL es un motor de base de datos **de alto rendimiento** que maneja cargas de trabajo complejas con facilidad, optimizando tanto la consulta como el procesamiento de datos. 🚀



## Seguridad de Nivel Empresarial

Con sólidas funciones de seguridad, PostgreSQL protege tus datos con autenticación avanzada, cifrado y control total sobre los accesos. 🔒

## Completamente Gratuito

PostgreSQL es una **herramienta de código abierto** y gratuita, sin costo adicional para comenzar a trabajar y desarrollar soluciones poderosas.



## Administración Simplificada

A través de su flexible y potente suite de herramientas, PostgreSQL permite una gestión eficiente y automatizada de bases de datos, maximizando tu productividad. 🛠️

## Multi-Plataforma

PostgreSQL es compatible con todos los sistemas operativos (Windows, macOS, Linux), sin costo adicional para comenzar a trabajar y desarrollar soluciones poderosas.



# Conectarse al motor de base de datos

1

## Abrir pgAdmin 4

-  Inicia pgAdmin 4 desde tu computadora.


2

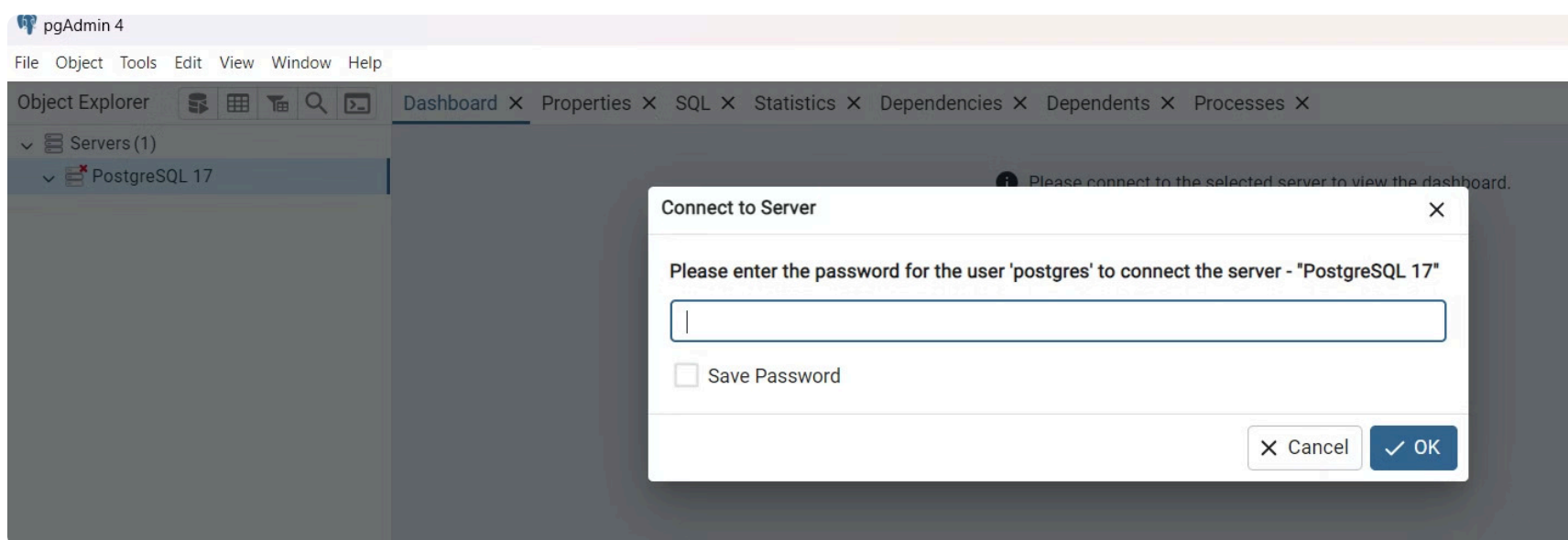
## Conectarse al servidor por defecto

- En el panel izquierdo, busca **"Servers"**. Haz clic en la flecha para expandirlo.
- Selecciona **"PostgreSQL [versión]"** (este es el servidor por defecto creado durante la instalación).

3





## Proveer credenciales de autenticación

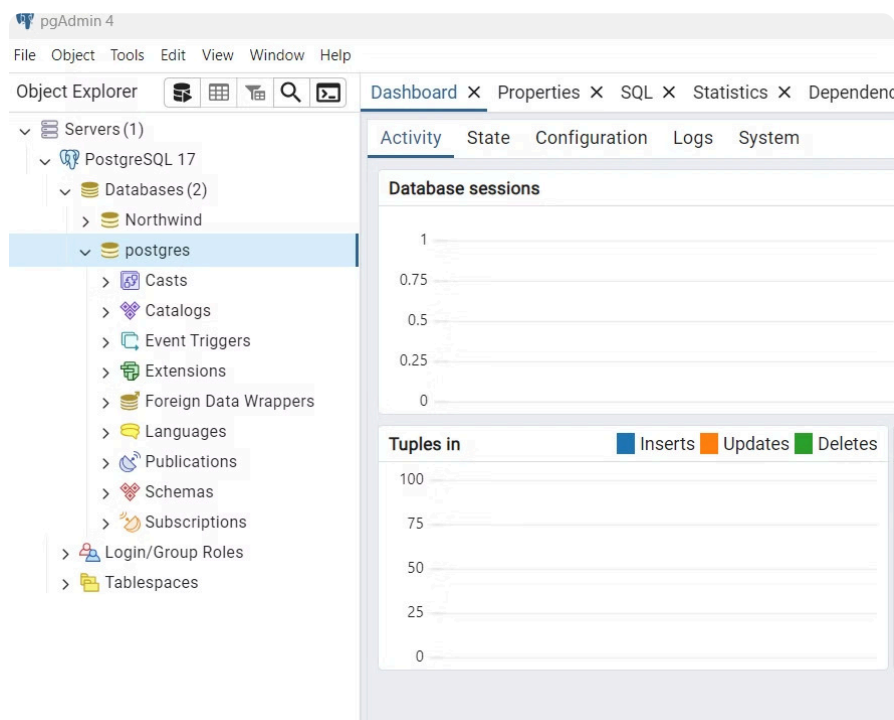
-  Si es la primera vez que accedes, pgAdmin pedirá una contraseña.
- Ingresa la contraseña que configuraste durante la instalación de PostgreSQL.
- Haz clic en **"OK"**.



4

## Verificar la conexión

-  Si la conexión es exitosa, el servidor se expandirá y verás:
  - Bases de datos 
  - Roles 
  - Tablas 
- Puedes expandir la base de datos **"postgres"** (base de datos de mantenimiento por defecto) para explorarla.

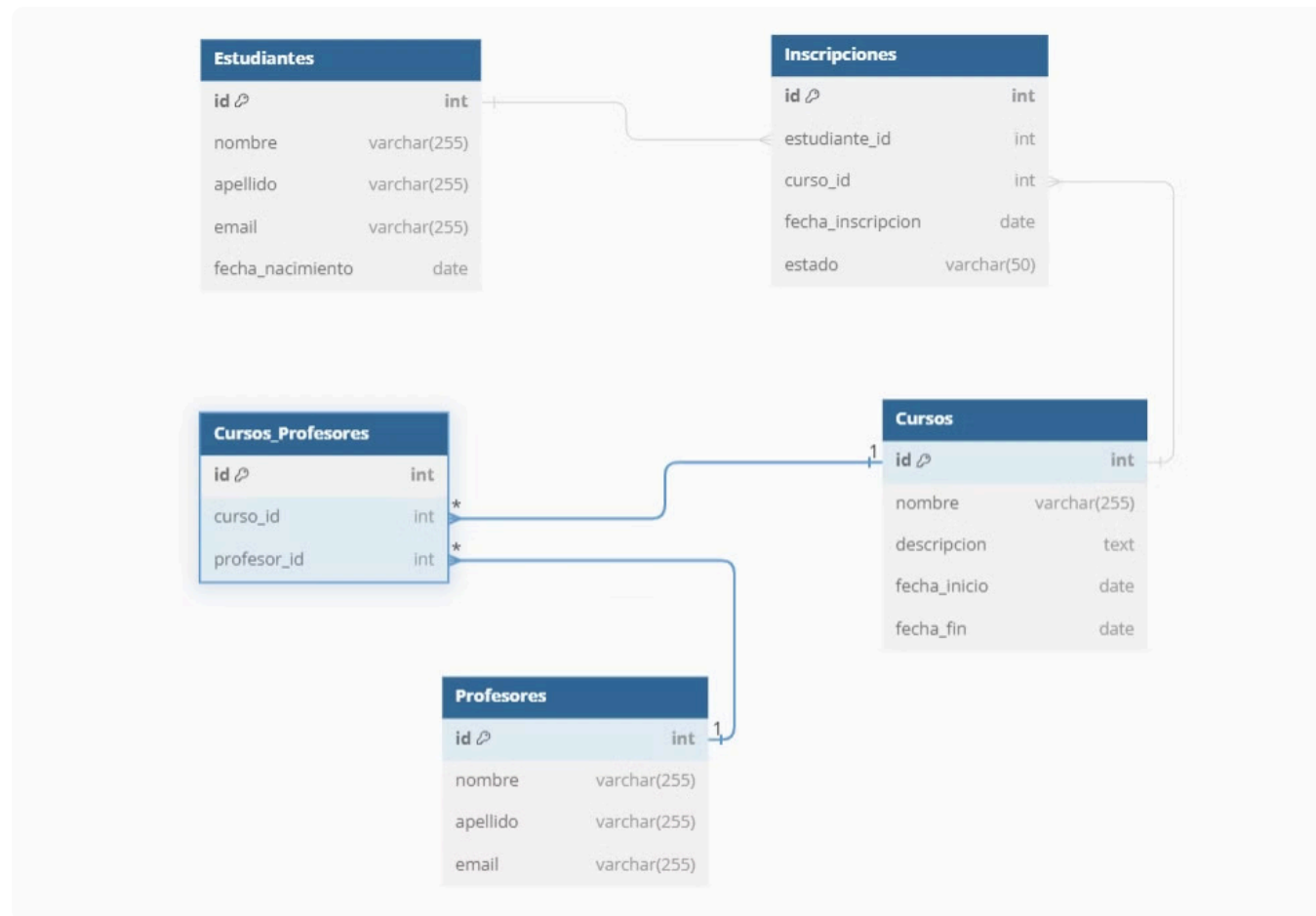


# Tablas en Bases de Datos

En una base de datos, una tabla es una estructura que organiza los datos en filas y columnas. Cada tabla representa una entidad o un conjunto de entidades relacionadas en un sistema de base de datos relacional.

**Base de datos:** ESCUELA

**Tablas:** Estudiantes, Inscripciones, Cursos, Profesores, Cursos\_Profesores



[https://dbdiagram.io/d/DIPLOMADOSQL\\_EJEM1-66a055ed8b4bb5230e2a9302](https://dbdiagram.io/d/DIPLOMADOSQL_EJEM1-66a055ed8b4bb5230e2a9302)



# PostgreSQL - Tipos de datos

Categoría	Tipos	Ejemplos
Text	char, varchar, text	char(10) - 'ABCDEFGHJIJ' varchar(50) - 'This is a string up to 50 characters' text - 'This is a longer string of text'
Numeric (Exact)	bigint, integer, smallint, decimal, numeric, money	bigint - 9223372036854775807 integer - 2147483647 decimal(10,2) - 123.45 money - '9,999,999,999.99'
Numeric (Approximate)	real, double precision	real - 3.402823E+38 double precision - 3.14159265358979
Date/Time	timestamp, date, time	timestamp - '2023-04-06 13:45:00' date - '2023-04-06' time - '13:45:00'
Binary	bytea	bytea - \x48656C6C6F (representación hexadecimal)
Other	json, jsonb, uuid, interval, inet, cidr, macaddr, point, line, box, circle, polygon	json - {"name": "John", "age": 30} uuid - '6F9619FF-8B86-D011-B42D-00C04FC964FF' interval - '1 day'

# Comandos básicos

1

## CREATE DATABASE

Crea una nueva base de datos.

Ejemplo: CREATE DATABASE MiBD;

2

## CREATE TABLE

Crea una nueva tabla en la base de datos.

Ejemplo: CREATE TABLE Clientes ( ClienteID SERIAL PRIMARY KEY, Nombre VARCHAR(50), Telefono VARCHAR(20) );

3

## CREATE SCHEMA

Crea un nuevo esquema en la base de datos.

Ejemplo: CREATE SCHEMA Ventas;

4

## DROP TABLE

Elimina una tabla existente.

Ejemplo: DROP TABLE Clientes;

5

## INSERT

Inserta nuevos datos en una tabla.

Ejemplo: INSERT INTO Clientes (ClienteID, Nombre, Telefono) VALUES (1, 'Juan Pérez', '555-1234');

6

## SELECT

Consulta datos de una tabla.

Ejemplo: SELECT Nombre, Telefono FROM Clientes;

7

## ALTER

Modifica la estructura de una tabla o base de datos.

Ejemplo: ALTER TABLE Clientes ADD COLUMN Email VARCHAR(50);

# Conexión a PostgreSQL con Python usando psycopg2

Para conectarte a una base de datos PostgreSQL desde Python, puedes usar el paquete **psycopg2**.

## 1. Instalación de psycopg2

Primero, necesitas instalar el paquete **psycopg2**. Puedes hacerlo usando **pip**:

```
pip install psycopg2
```

## 2. Código para conectarse a la base de datos

```
import psycopg2
import os

# Establecer conexión con la base de datos
def conectar_a_base_de_datos():
    try:
        conn = psycopg2.connect(
            host="localhost", # Cambia si es necesario
            database="Northwind", # Nombre de tu base de datos
            user="postgres", # Usuario
            password=os.getenv('PASSWORD_PSQL'), # Contraseña desde variable de entorno
            port=5432 # Puerto por defecto
        )
        print("Conexión exitosa a la base de datos.")
        return conn
    except Exception as e:
        print(f"Error al conectar: {e}")
        return None

# Llamar a la función de conexión
conn = conectar_a_base_de_datos()
```