

# Assignment 1

Due by **September 26th, 2025**

Total Points: 50 Marks

Course Grade Weight: 11%

## Collaboration Policy

This assignment can be completed in pairs or individually. However, both team members must be fully familiar with all parts of the submission, including the database setup, Python scripts, SQL queries, and written explanations.

**Important:** During grading or review, if either team member is unable to explain any part of the work, full marks for that part will not be gone. You are expected to collaborate closely and understand the entire solution together.

## Grading

You need to have everything requested in this assignment in your submission and demo it to TAs when announced. You will be required to run the code to verify it's working.

## ChatGPT

Do not ask ChatGPT (Or other AI tools) to generate the SQL queries. The purpose of this assignment is that you learn how to craft the queries yourself. That is still a fundamental skill you need to have for a while.

## Deliverables

1. Screenshots of the parts where the screenshots are required. Please name them clearly.
2. All the code (Python, SQLs). Also, name them clearly.
3. A simple short Readme file that shows what file is related to what part.

# Overview

You will build a simple PostgreSQL-based database system for a store. The database includes information about customers, orders, and deliveries. You will:

1. Create the schema and load data from provided CSV files
2. Use Python to interact with the database via dynamic queries and data manipulation functions
3. Write SQL queries to extract insights from the database

Note: Unless something is explicitly requested in the assignment, you're free to do it any way you'd like.

## Files

The following files are uploaded as the initial data.

- Customers.csv
- Orders.csv
- deliveries.csv

## Part 1: Set Up PostgreSQL Server Instance (10 Marks)

Install and run a PostgreSQL database server on your local machine so you can connect to it via command line, Python and IDE (5 marks). Find an IDE (e.g., DataGrip) that allows you to connect to your database server (2 marks). Create a database named **storedb** and run the below code to create the main tables. (3 marks)

```
CREATE TABLE customers (  
    customer_id SERIAL PRIMARY KEY  
    ,NAME TEXT  
    ,email TEXT  
    ,phone TEXT  
    ,address TEXT  
);  
  
CREATE TABLE orders (  
    order_id SERIAL PRIMARY KEY  
    ,customer_id INT REFERENCES customers(customer_id)  
    ,order_date DATE
```

```
,total_amount NUMERIC
,product_id INT
,product_category TEXT
,product_name TEXT
);

CREATE TABLE deliveries (
    delivery_id SERIAL PRIMARY KEY
,order_id INT REFERENCES orders(order_id)
,delivery_date DATE
,STATUS TEXT
);
```

Take a screenshot that shows the above tables are created.

## Part 2: Write a Python Script to Load CSV Data into PostgreSQL (5 Marks)

Write a Python script named **load.py** that connects to the storedb PostgreSQL database using psycopg2 connector and loads data from the provided customers.csv, orders.csv, and deliveries.csv files into their respective tables by reading each CSV and inserting the rows into the database.

Take screenshots of the content of each table and include it in your submission.

## Part 3: Add and Update Data with Python Functions (10 Marks)

Extend your Python script by implementing the following functions that allow you to add new records or update existing ones in the database. Write one function for each of the following.

1. Add a new customer with attributes like name, email, phone, and address.
2. Add a new order with attributes including customer ID, order date, total amount, product ID, product category, and product name.
3. Add a new delivery with attributes such as order ID, delivery date, and status.
4. Update the delivery status for an existing delivery, given its delivery ID.

Then, using the functions above, perform the below operations.

1. Add customer: Liam Nelson, liam.nelson@example.com, 555-2468, 111 Elm Street.
2. Add order for Liam (use returned customer\_id): 2025-06-01, 180.00, 116, Electronics, Bluetooth Speaker.
3. Add delivery for this order (use returned order\_id): 2025-06-03, Pending.
4. Update this delivery's status to Shipped (use returned delivery\_id).
5. Add one more customer, order, and delivery with any valid data.
6. Update delivery status of delivery\_id = 3 to Delivered.

Take screenshots of the content of each table after performing all the steps above and include it in your submission. Then, continue to step 4. (Queries in Step 4 should be executed on the updated data after this step)

## Part 4: SQL Queries (15 Marks)

Write SQL queries to retrieve the requested information from the database tables. You don't have to do this in Python, you can just directly run your queries against the database using IDE but you should store the SQL code in a file you submit and also include a screenshot of the output of each part. Please use clear names for the SQL files and screenshot files (one file and screenshot per query).

Use your judgment to decide which columns to include in each query's output. For example, query 6 might include the product category and the count of orders, while query 7 could show customer details along with their number of orders. Your selected columns should clearly represent the results, but there is no single correct answer.

1. List all customers with their contact details. (1 Mark)
2. Count the total number of orders in the database. (1 Mark)
3. Find all orders made by customer "Alice Johnson". (2 Marks)
4. List all **orders** that have not yet been delivered (delivery status not "Delivered"). (2 Marks)
5. Find the total amount spent by each customer, sorted from highest to lowest. (2 Marks)
6. Find the number of orders per product category. (2 Marks)
7. Find customers who have placed more than 2 orders. (2 Marks)
8. Find the product category with the highest total sales amount. (3 Marks)

## Part 5: Expanding the Schema (10 Marks)

What other tables could have been included in the design to make a more complete database for the store?

Introduce two useful tables, describe a list of their attributes and show how they would relate to the existing schema. For each table, write its CREATE TABLE statement and describe what changes (foreign keys or new columns) would need to be made to the existing tables.

Hint: Payments can be one of the answers.

Good Luck!