# SQL Mini-Project Exercises

Use the following schema for a retail business scenario to complete the exercises below:  
- Customers (CustomerID, FirstName, LastName, Email, Phone)  
- Products (ProductID, ProductName, Price, Category)  
- Orders (OrderID, CustomerID, OrderDate, ProductID, Quantity)

### Dataset Details:

**1. Customers Table:**

* CustomerID: Unique identifier for each customer.
* FirstName: Customer's first name.
* LastName: Customer's last name.
* Email: Customer's email address.
* Phone: Customer's phone number.

**2. Products Table:**

* ProductID: Unique identifier for each product.
* ProductName: Name of the product.
* Price: Price of the product.
* Category: Category of the product.

**3. Orders Table:**

* OrderID: Unique identifier for each order.
* CustomerID: Identifier linking the order to a customer.
* OrderDate: The date the order was placed.
* ProductID: Identifier linking the order to a product.
* Quantity: Number of units ordered.

Please download these CSV file in your local folder and then import that in your SQL Server Management studio or MySQL (Whichever tool you are using).

<https://github.com/AtulKadlag/Mini_projects_dataAnalyst/blob/main/Customers.csv>

<https://github.com/AtulKadlag/Mini_projects_dataAnalyst/blob/main/Orders.csv>

<https://github.com/AtulKadlag/Mini_projects_dataAnalyst/blob/main/Products.csv>

Once the data is imported successfully, you will have three tables in your SQL tool, then solve below questions. Please write SQL queries for each question.

## Basic SQL Queries

Retrieve all customer information.

SQL Query:

List all products under the category 'Electronics'.

SQL Query:

## SQL Statement Types

Add a new customer to the Customers table.

SQL Query:

Update the phone number for a specific customer.

SQL Query:

Delete a customer from the database based on CustomerID.

SQL Query:

## Joins

Display the full order details along with customer information.

SQL Query:

Show all products that have never been ordered.

SQL Query:

## GROUP BY and Aggregation

Find the total number of products sold for each category.

SQL Query:

Determine the average price of products in each category.

SQL Query:

## ORDER BY

List all customers sorted by last name in ascending order.

SQL Query:

Display all orders from the most recent to the oldest.

SQL Query: