Lab 4

- Create a function that returns the total number of orders made by a specific customer using their customer_id.
- Write a function that takes a product_id and returns the category name of the product.
- Create a procedure that adds a new customer with their full details (name, contact name, address, etc.).
- Write a procedure that updates the category of a given product by its product_id.
- Create a trigger that prevents deleting a category if there are still products assigned to it.
- Create a view that displays all products with their price and a label ("Cheap", "Moderate", "Expensive") depending on the price value.
- Create a view that shows each customer with the number of orders they have made.
- Which index would you create to speed up filtering orders by customer_id
- Create a unique index on the category_name column to prevent duplicate category names.
- Write a transaction that inserts a new customer and a new order for that customer. If either fails, roll back.
- Write a subquery to find all customers who have placed orders for products in a specific category (use category_id as input).
- Create a function that calculates the total revenue generated from a specific product (based on product_id) across all orders.
- Create a view that lists the top 5 customers based on the total quantity of products ordered.

- Write a query using a subquery to find products that have never been ordered.
- Create a user role with read-only access to the customers and orders tables, and grant it to a new user then revoke its privileges .
- Create a backup for the database.
- Restore the database.