1. Hands-on Exercise: Filtering Data using SQL Queries

Retrieve all products from the Products table that belong to the category 'Electronics' and have a price greater than 500.

2. Hands-on Exercise: Total Aggregations using SQL Queries

Calculate the total quantity of products sold from the Orders table.

3. Hands-on Exercise: Group By Aggregations using SQL Queries

Calculate the total revenue generated for each product in the Orders table.

4. Hands-on Exercise: Order of Execution of SQL Queries

Write a query that uses WHERE, GROUP BY, HAVING, and ORDER BY clauses and explain the order of execution.

5. Hands-on Exercise: Rules and Restrictions to Group and Filter Data in SQL Queries

Write a query that corrects a violation of using non-aggregated columns without grouping them.

6. Hands-on Exercise: Filter Data based on Aggregated Results using Group By and Having

Retrieve all customers who have placed more than 5 orders using GROUP BY and HAVING clauses.

1. Basic Stored Procedure

Create a stored procedure named GetAllCustomers that retrieves all customer details from the Customers table.

2. Stored Procedure with Input Parameter

Create a stored procedure named GetOrderDetailsByOrderID that accepts an OrderID as a parameter and retrieves the order details for that specific order.

3. Stored Procedure with Multiple Input Parameters

Create a stored procedure named GetProductsByCategoryAndPrice that accepts a product Category and a minimum Price as input parameters and retrieves all products that meet the criteria.

4. Stored Procedure with Insert Operation

Create a stored procedure named InsertNewProduct that accepts parameters for ProductName, Category, Price, and StockQuantity and inserts a new product into the Products table.

5. Stored Procedure with Update Operation

Create a stored procedure named UpdateCustomerEmail that accepts a CustomerID and a NewEmail parameter and updates the email address for the specified customer.

6. Stored Procedure with Delete Operation

Create a stored procedure named DeleteOrderByID that accepts an OrderID as a parameter and deletes the corresponding order from the Orders table.

7. Stored Procedure with Output Parameter

Create a stored procedure named GetTotalProductsInCategory that accepts a Category parameter and returns the total number of products in that category using an output parameter.