ADBMS laboratory work:

1. Design the database using DDL command to define the following relational schema **[RDBMS].**

   ProductCategory(CategoryId, CategoryName)
   Product (ProductId, Name, Quantity, Discount, Price, ProductCategoryRef_Id, CreatedDate, ProductCode)

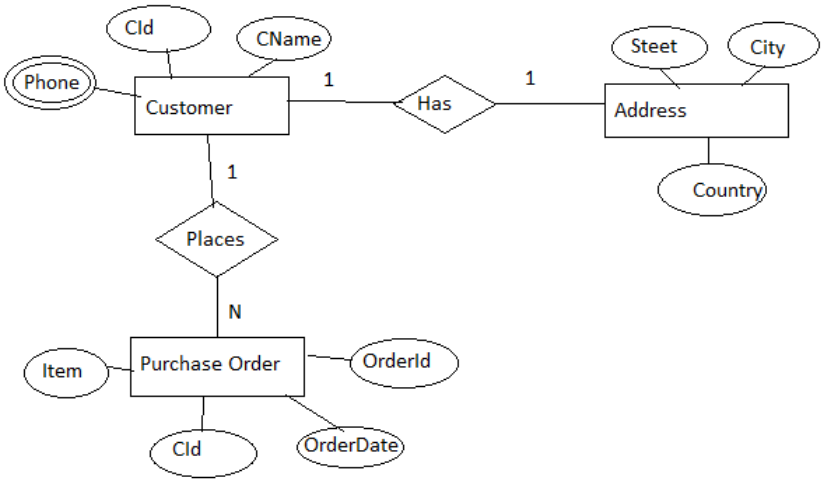
   Column constraints: NOT NULL on Name, CHECK on Price, Foreign key on ProductCategoryRef_Id, UNIQUE on ProductCode)

   Address (AddressId, City, Street, CountryId)
   Customer (CustomerId, AddressRef_Id, Name, Address, EmailAddress, PhoneNumber)
   PurchaseOrder (PurchaseOrderId, ProductRef_Id, CustomerRef_Id, OrderDate, PaymentTotal, OrderQuantity)
   Country (CountryId, Name)

   and perform the following queries,

   - Insert records with your own value into all relations.
   - Find the maximum price, the minimum price, and the average price among all products (Aggregate functions).
   - For each product category, retrieve the categoryId, the number of products in the product category, and their total price (Grouping).
   - Write a SQL statement to know which product name and product category on which product belongs (inner join).
   - Get product name, product price and product category, order by product name from "product" and "productcategory" for all product even they have not assigned to any category (Left Outer join)).
   - Write a query to create a view that shows the total amount and total orders for each customer.

2. Create procedure that accepts a markup amount and categoryId as input, applies the markup to the price, and returns the total number of items of the specific category **[RDBMS].**
3. Create an SQL function that calculates the percentage of sales for each product category out of the total sales **[RDBMS].**
4. Design the object relational database (using ODL) for the following ER diagram **[ORDBMS].**



5. Implement row-level trigger and statement-level trigger in PL/SQL **[Active database concept]**.
6. Implement temporal tables (system-versioned) in SQL **[Temporal Database Concepts].**
7. Write a spatial SQL query to perform a range and nearest neighbor query in SQL Server **[Spatial Database].**
8. Consider the following set of facts for the relation supervise (X, Y), where Y is the supervisor of X **[Deductive Database Concepts]:**

> supervise(james,franklin).

> supervise(franklin,john).

> supervise(franklin,ramesh).

> supervise(franklin,joyce).

> supervise(james,jennifer).

> supervise(jennifer,alicia).

> supervise(jennifer,ahmed).

Consider the rules

> superior(X,Y):-supervise(X,Y).

> superior(X,Y):-supervise(X,Z),superior(Z,Y).

Perform the following queries:

supervise(james,franklin)?

superior(james,franklin)?

superior(X,Y)?

superior(franklin,X)?

superior(X,ram)?

9. Basic query operations in MongoDB **[NOSQL Systems].**