New lab 4

Friday, August 30, 2024   11:22 AM

Certainly! Here's a table format that represents the entities (Products and Customers) and their attributes, along with some example data.
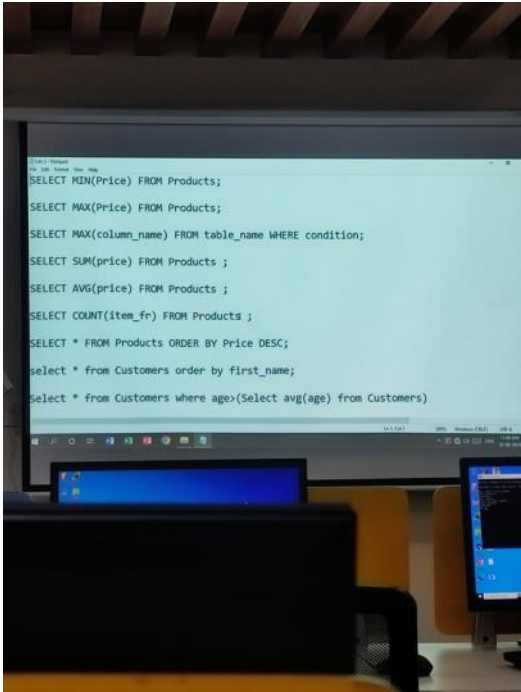
Entity: Products

ProductID	ItemName	Price	Category	StockQuantity
1	Laptop	800	Electronics	50
2	Headphones	50	Electronics	150
3	Coffee Maker	120	Home	75
4	Smartphone	600	Electronics	100
5	Desk Lamp	30	Home	200

Entity: Customers

CustomerID	First_Name	Last_Name	Age	Email	City
1	John	Doe	28	john.doe@example.com	New York
2	Jane	Smith	34	jane.smith@example.com	Los Angeles
3	Alice	Johnson	40	alice.j@example.com	Chicago
4	Bob	Brown	23	bob.brown@example.com	Houston
5	Carol	Davis	30	carol.d@example.com	Miami

Example SQL Queries Using These Tables:


```
INSERT INTO Products (ProductID, ItemName, Price, Category, StockQuantity) VALUES( (1, 'Laptop', 800, 'Electronic', 50), (2, 'Headphones', 50, 'Electronics', 150), (3, 'Coffee Maker', 120, 'Home', 75), (4, 'Smartphone', 600, 'Electronics', 100), (5, 'Desk Lamp', 30, ' Home', 200));
```

```
-- Step 1: Create the Products table CREATE TABLE Products ( ProductID INT PRIMARY KEY, ItemName VARCHAR(50), Price DECIMAL(10, 2), Category VARCHAR(50), StockQuantity INT ); -- Step 2: Insert data into the Products table INSERT INTO Products (ProductID, ItemName, Price, Category, StockQuantity) VALUES (1, 'Laptop', 800.00, 'Electronics', 50), (2, 'Headphones', 50.00, 'Electronics', 150), (3, 'Coffee Maker', 120.00, 'Home', 75), (4, 'Smartphone', 600.00, 'Electronics', 100), (5, 'Desk Lamp', 30.00, 'Home', 200);
```

```
CREATE TABLE Products ( ProductID INT PRIMARY KEY, ItemName VARCHAR(50), Price DECIMAL(10, 2), Category VARCHAR(50), StockQuantity INT );
```

```
INSERT INTO Products (ProductID, ItemName, Price, Category, StockQuantity) VALUES (1, 'Laptop', 800, 'Electronics', 50);
```

```
INSERT INTO Products1 (ProductID, ItemName, Price, Category, StockQuantity) VALUES (2, 'Headphones', 50, 'Electronics', 150); INSERT INTO Products1 (ProductID, ItemName, Price, Category, StockQuantity) VALUES (3, 'Coffee Maker', 120, 'Home', 75); INSERT INTO Products1 (ProductID, ItemName, Price, Category, StockQuantity) VALUES (4, 'Smartphone', 600, 'Electronics', 100); INSERT INTO Products1 (ProductID, ItemName, Price, Category, StockQuantity) VALUES (5, 'Desk Lamp', 30, 'Home', 200);
```

```
CREATE TABLE Customers ( CustomerID NUMBER PRIMARY KEY, First_Name VARCHAR2(15), Last_Name VARCHAR2(10), Age NUMBER, Email VARCHAR2(20), City VARCHAR2(10) );
```

INSERT INTO Customers (CustomerID, First_Name, Last_Name, Age, Email, City) VALUES (1, 'John', 'Doe', 28, 'john.doe@example.com', 'New York');

INSERT INTO Customers (CustomerID, First_Name, Last_Name, Age, Email, City) VALUES (2, 'Jane', 'Smith', 34, 'jane.smith@example.com', 'Los Angeles'); INSERT INTO Customers (CustomerID, First_Name, Last_Name, Age, Email, City) VALUES (3, 'Alice', 'Johnson', 40, 'alice.j@example.com', 'Chicago'); INSERT INTO Customers (CustomerID, First_Name, Last_Name, Age, Email, City) VALUES (4, 'Bob', 'Brown', 23, 'bob.brown@example.com', 'Houston'); INSERT INTO Customers (CustomerID, First_Name, Last_Name, Age, Email, City) VALUES (5, 'Carol', 'Davis', 30, 'carol.d@example.com', 'Miami');
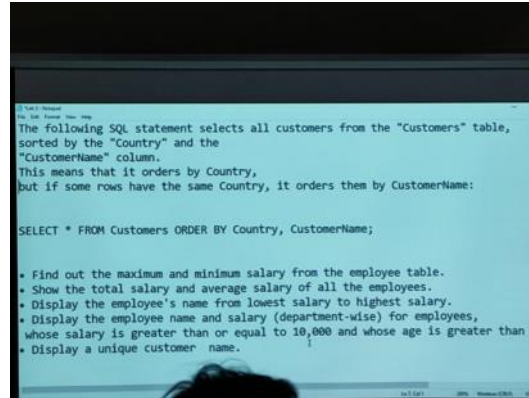
Certainly! Here's how you can create the Employees and Customers tables with appropriate columns for the SQL queries you've outlined.

Creating the Employees Table

sql
Copy code
```
-- Create the Employees table
CREATE TABLE Employees (
    EmployeeID NUMBER PRIMARY KEY,
    EmployeeName VARCHAR2(50),
    Salary NUMBER(10, 2),
    Age NUMBER,
    Department VARCHAR2(50)
);
```

Creating the Customers Table

sql
Copy code
```
-- Create the Customers table
CREATE TABLE Customers (
    CustomerID NUMBER PRIMARY KEY,
    CustomerName VARCHAR2(50),
    Country VARCHAR2(50)
);
```


The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column.
This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName:

SELECT * FROM Customers ORDER BY Country, CustomerName;

- Find out the maximum and minimum salary from the employee table.
- Show the total salary and average salary of all the employees.
- Display the employee's name from lowest salary to highest salary.
- Display the employee name and salary (department-wise) for employees, whose salary is greater than or equal to 10,000 and whose age is greater than 25
- Display a unique customer name.

Inserting Sample Data into the Employees Table

sql
Copy code
```
-- Insert sample data into Employees table
INSERT INTO Employees (EmployeeID, EmployeeName, Salary, Age, Department) VALUES (1, 'Alice Johnson', 9500, 30, 'HR');
INSERT INTO Employees (EmployeeID, EmployeeName, Salary, Age, Department) VALUES (2, 'Bob Brown', 12000, 28, 'IT');
INSERT INTO Employees (EmployeeID, EmployeeName, Salary, Age, Department) VALUES (3, 'Carol Davis', 15000, 35, 'IT');
INSERT INTO Employees (EmployeeID, EmployeeName, Salary, Age, Department) VALUES (4, 'David Smith', 9000, 40, 'Finance');
INSERT INTO Employees (EmployeeID, EmployeeName, Salary, Age, Department) VALUES (5, 'Eva Green', 11000, 26, 'HR');
```

Inserting Sample Data into the Customers Table

sql
Copy code
```
-- Insert sample data into Customers table
INSERT INTO Customers (CustomerID, CustomerName, Country) VALUES (1, 'John Doe', 'USA');
INSERT INTO Customers (CustomerID, CustomerName, Country) VALUES (2, 'Jane Smith', 'Canada');
INSERT INTO Customers (CustomerID, CustomerName, Country) VALUES (3, 'Alice Johnson', 'USA');
INSERT INTO Customers (CustomerID, CustomerName, Country) VALUES (4, 'Bob Brown', 'UK');
INSERT INTO Customers (CustomerID, CustomerName, Country) VALUES (5, 'Carol Davis', 'Canada');
```

CREATE TABLE Employees (EmployeeID NUMBER PRIMARY KEY, EmployeeName VARCHAR2(15),Salary NUMBER(10, 2), Age NUMBER,Department VARCHAR2(15));