--Create the Employees Table:--

Write an SQL query to create the "Employees" table with the following columns:

EmployeeID (integer)

FirstName (text)

LastName (text)

Department (text)

Salary (decimal)

---------------------------- table created -------------------------

Create table Employees(

EmployeeID INT PRIMARY KEY,

FirstName text,

LastName text,

Department text,

Salary decimal (10, 2)

);

--------------------------- record inserted --------------------------

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(1,"John","Doe","HR",50000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(2,"Jane","Smith","HR",55000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(3,"Bob","Johnson","IT",60000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(4,"Alice","Williams","IT",65000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(5,"David","Brown","IT",62000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(6,"Mary","Davis","Finance",58000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(7,"Tom","Wilson","Finance",60000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(8,"Linda","Lee","Marketing",54000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(9,"Mike","Clark","Marketing",56000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(10,"Emily","Thomas","IT",63000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(11,"Peter","Evans","Finance",59000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(12,"Sara","Martin","Marketing",55000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(13,"Chris","Roberts","IT",61000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(14,"Laura","Garcia","HR",52000);

INSERT INTO Employees("EmployeeID","FirstName","LastName","Department","Salary") VALUES(15,"Mark","Harris","Finance",57000);

15 rows inserted.

Select \* from Employees;

EmployeeID FirstName LastName Department Salary

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 John Doe HR 50000

2 Jane Smith HR 55000

3 Bob Johnson IT 60000

4 Alice Williams IT 65000

5 David Brown IT 62000

6 Mary Davis Finance 58000

7 Tom Wilson Finance 60000

8 Linda Lee Marketing 54000

9 Mike Clark Marketing 56000

10 Emily Thomas IT 63000

11 Peter Evans Finance 59000

12 Sara Martin Marketing 55000

13 Chris Roberts IT 61000

14 Laura Garcia HR 52000

15 Mark Harris Finance 57000

--Create the Products Table:--

Write an SQL query to create the "Products" table with the following columns:

ProductID (integer)

ProductName (text)

Category (text)

Price (decimal)

StockQuantity (integer)

---------------------------- table created --------------------------

CREATE TABLE Products (

ProductID INT,

ProductName TEXT,

Category TEXT,

Price DECIMAL(10, 2),

StockQuantity INT

);

INSERT INTO Product VALUES(1,"Laptop","Electronics",800.00,10);

INSERT INTO Product VALUES(2,"Smartphone","Electronics",500.00,15);

INSERT INTO Product VALUES(3,"Desk Chair","Furniture",150.00,20);

INSERT INTO Product VALUES(4,"Coffee Table","Furniture",200.00,10);

INSERT INTO Product VALUES(5,"Printer","Electronics",100.00,5);

INSERT INTO Product VALUES(6,"Sofa","Furniture",450.00,12);

INSERT INTO Product VALUES(7,"T-shirt","Clothing",200.00,50);

INSERT INTO Product VALUES(8,"Jeans","Clothing",40.00,30);

INSERT INTO Product VALUES(9,"Microwave","Appliances",120.00,8);

INSERT INTO Product VALUES(10,"Refrigerator","Appliances",600.00,5);

INSERT INTO Product VALUES(11,"Dining Table","Furniture",350.00,8);

INSERT INTO Product VALUES(12,"Headphones","Electronics",60.00,25);

INSERT INTO Product VALUES(13,"Shoes","Clothing",70.00,40);

INSERT INTO Product VALUES(14,"Blender","Appliances",50.00,10);

INSERT INTO Product VALUES(15,"TV","Electronics",900.00,6);

15 rows inserted.

Select \* from Products;

ProductID ProductName Category Price StockQuantity

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 Laptop Electronics 800 10

2 Smartphone Electronics 500 15

3 Desk Chair Furniture 150 20

4 CoffeeTable Furniture 200 10

5 Printer Electronics 100 5

6 Sofa Furniture 450 12

7 T-shirt Clothing 200 50

8 Jeans Clothing 40 30

9 Microwave Appliances 120 8

10 Refrigerator Appliances 600 5

11 DiningTable Furniture 350 8

12 Headphones Electronics 60 25

13 Shoes Clothing 70 40

14 Blender Appliances 50 10

15 TV Electronics 900 6

--Create the Orders Table:--

Write an SQL query to create the "Orders" table with the following columns:

OrderID (integer)

CustomerName (text)

OrderDate (date)

TotalAmount (decimal)

---------------------------- table created --------------------------

CREATE TABLE Orders (

OrderID INT,

CustomerName TEXT,

OrderDate DATE,

TotalAmount DECIMAL(10, 2)

);

INSERT INTO Orders VALUES(1,"John Smith",2023-10-15,300);

INSERT INTO Orders VALUES(2,"Jane Doe",2023-10-16,450);

INSERT INTO Orders VALUES(3,"Bob Johnson",2023-10-17,600);

INSERT INTO Orders VALUES(4,"Alice Williams",2023-10-18,750);

INSERT INTO Orders VALUES(5,"David Brown",2023-10-19,400);

INSERT INTO Orders VALUES(6,"Mary Davis",2023-10-20,550);

INSERT INTO Orders VALUES(7,"Tom Wilson",2023-10-21,700);

INSERT INTO Orders VALUES(8,"Linda Lee",2023-10-22,350);

INSERT INTO Orders VALUES(9,"Mike Clark",2023-10-23,900);

INSERT INTO Orders VALUES(10,"Emily Thomas",2023-10-24,200);

INSERT INTO Orders VALUES(11,"Peter Evans",2023-10-25,450);

INSERT INTO Orders VALUES(12,"Sara Martin",2023-10-26,800);

INSERT INTO Orders VALUES(13,"Chris Roberts",2023-10-27,350);

INSERT INTO Orders VALUES(14,"Laura Garcia",2023-10-28,600);

INSERT INTO Orders VALUES(15,"Mark Harris",2023-10-29,950);

15 rows inserted.

Select \* from Orders;

OrderID CustomerName OrderDate TotalAmount

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

1 John Smith 2023-10-15 300

2 Jane Doe 2023-10-16 450

3 Bob Johnson 2023-10-17 600

4 Alice Williams 2023-10-18 750

5 David Brown 2023-10-19 400

6 Mary Davis 2023-10-20 550

7 Tom Wilson 2023-10-21 700

8 Linda Lee 2023-10-22 350

9 Mike Clark 2023-10-23 900

10 Emily Thomas 2023-10-24 200

11 Peter Evans 2023-10-25 450

12 Sara Martin 2023-10-26 800

13 Chris Roberts 2023-10-27 350

14 Laura Garcia 2023-10-28 600

15 Mark Harris 2023-10-29 950

--Create the Customers Table:--

Write an SQL query to create the "Customers" table with the following columns:

CustomerID (integer)

CustomerName (text)

City (text)

State (text)

---------------------------- table created --------------------------

CREATE TABLE Customers (

CustomerID INT,

CustomerName TEXT,

City TEXT,

State TEXT

);

INSERT INTO Customers VALUES(1,"John Smith","New York","NY");

INSERT INTO Customers VALUES(2,"Jane Doe","Los Angeles","CA");

INSERT INTO Customers VALUES(3,"Bob Johnson","Chicago","IL");

INSERT INTO Customers VALUES(4,"Alice Williams","Houston","TX");

INSERT INTO Customers VALUES(5,"David Brown","Philadelphia","PA");

INSERT INTO Customers VALUES(6,"Mary Davis","Phoenix","AZ");

INSERT INTO Customers VALUES(7,"Tom Wilson","San Antonio","TX");

INSERT INTO Customers VALUES(8,"Linda Lee","San Diego","CA");

INSERT INTO Customers VALUES(9,"Mike Clark","Dallas","TX");

INSERT INTO Customers VALUES(10,"Emily Thomas","Austin","TX");

INSERT INTO Customers VALUES(11,"Peter Evans","San Francisco","CA");

INSERT INTO Customers VALUES(12,"Sara Martin","Seattle","WA");

INSERT INTO Customers VALUES(13,"Chris Roberts","Denver","CO");

INSERT INTO Customers VALUES(14,"Laura Garcia","Boston","MA");

INSERT INTO Customers VALUES(15,"Mark Harris","Miami","FL");

15 rows inserted.

Select \* from Customers;

CustomerID CustomerName City State

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

1 John Smith New York NY

2 Jane Doe Los Angeles CA

3 Bob Johnson Chicago IL

4 Alice Williams Houston TX

5 David Brown Philadelphia PA

6 Mary Davis Phoenix AZ

7 Tom Wilson San Antonio TX

8 Linda Lee San Diego CA

9 Mike Clark Dallas TX

10 Emily Thomas Austin TX

11 Peter Evans San Francisco CA

12 Sara Martin Seattle WA

13 Chris Roberts Denver CO

14 Laura Garcia Boston MA

15 Mark Harris Miami FL

----GROUP BY----

--For the Employees Table:---

1.)Find the average salary for all employees.

SELECT AVG(Salary) AS AverageSalary FROM Employees.

OUTPUT:=

AVERAGESALARY

-------------

57800

2.)List the departments and the total number of employees in each department.

SELECT Department, COUNT(\*) AS NumberOfEmployees FROM Employees GROUP BY Department;

OUTPUT:=

DEPARTMENT NUMBEROFEMPLOYEES

---------- -----------------

Marketing 3

HR 3

IT 5

Finance 4

3.)Calculate the total salary for the HR department.

SELECT Department, SUM(Salary) AS TotalSalary FROM Employees WHERE Department = 'HR' GROUP BY Department;

OUTPUT:=

TOTAL\_SALARY

------------

157000

4.)Find the department with the highest average salary.

SELECT Department, AVG(Salary) AS AverageSalary FROM Employees GROUP BY Department ORDER BY AverageSalary DESC

LIMIT 1;

OUTPUT:=

DEPARTMENT AVERAGESALARY

---------- --------------

IT 62200

Finance 58500

Marketing 55000

HR 52333.3333

5.)List the departments and the maximum salary in each department.

SELECT Department, MAX(Salary) AS MaximumSalary FROM Employees GROUP BY Department;

OUTPUT:=

DEPARTMENT MAXSALARY

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

Marketing 56000

HR 55000

IT 65000

Finance 60000

6.)Count the number of employees in the Marketing department.

SELECT COUNT(\*) AS NumberOfEmployees FROM Employees WHERE Department = 'Marketing';

OUTPUT:=

NUMBEROFEMPLOYEES

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3

7.)Find the employee with the highest salary.

SELECT \* FROM Employees ORDER BY Salary DESC LIMIT 1;

OUTPUT:=

EMPLOYEEID FIRSTNAME LASTNAME DEPARTMENT SALARY

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

4 Alice Williams IT 65000

10 Emily Thomas IT 63000

5 David Brown IT 62000

13 Chris Roberts IT 61000

3 Bob Johnson IT 60000

7 Tom Wilson Finance 60000

11 Peter Evans Finance 59000

6 Mary Davis Finance 58000

15 Mark Harris Finance 57000

9 Mike Clark Marketing 56000

12 Sara Martin Marketing 55000

2 Jane Smith HR 55000

8 Linda Lee Marketing 54000

14 Laura Garcia HR 52000

1 John Doe HR 50000

8.)List the employees in the IT department in alphabetical order by last name.

SELECT \* FROM Employees WHERE Department = 'IT' ORDER BY LastName;

OUTPUT:=

EMPLOYEEID FIRSTNAMELASTNAME DEPARTMENT SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 David Brown IT 62000

3 Bob Johnson IT 60000

13 Chris Roberts IT 61000

10 Emily Thomas IT 63000

4 Alice Williams IT 65000

--For the Products Table:--

1.)Calculate the average price for products in the "Electronics" category.

SELECT Category, AVG(Price) AS AveragePrice FROM Products WHERE Category = 'Electronics' GROUP BY Category;

OUTPUT:=

AVERAGE\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_

472

2.)List the categories and the total number of products in each category.

SELECT Category, COUNT(\*) AS TotalProducts FROM Products GROUP BY Category;

OUTPUT:=

CATEGORY TOTAL\_PRODUCTS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 5

Furniture 4

Clothing 3

Appliances 3

3.)Find the category with the highest average price.

SELECT Category, AVG(Price) AS AveragePrice FROM Products GROUP BY Category ORDER BY AveragePrice DESC

LIMIT 1;

OUTPUT:=

CATEGORY AVERAGE\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 390

Furniture 287.5

Appliances 256.666667

Clothing 43.3333333

4.)Calculate the total stock quantity for the "Furniture" category.

SELECT Category, SUM(StockQuantity) AS TotalStockQuantity FROM Products WHERE Category = 'Furniture' GROUP BY Category;

OUTPUT:=

CATEGORYTOTAL STOCK\_QUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Furniture 50

5.)List the categories and the minimum price in each category.

SELECT Category, MIN(Price) AS MinimumPrice FROM Products GROUP BY Category;

OUTPUT:=

CATEGORY MINIMUM\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 60

Furniture 150

Clothing 20

Appliances 50

6.)Count the number of products in the "Clothing" category.

SELECT COUNT(\*) AS NumberOfProducts FROM Products WHERE Category = 'Clothing';

OUTPUT:=

CATEGORY PRODUCT\_COUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Clothing 3

7.)Find the product with the highest price.

SELECT \* FROM Products ORDER BY Price DESC LIMIT 1;

OUTPUT:=

PRODUCTID PRODUCTNAME CATEGORY PRICE STOCKQUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15 TV Electronics 900 6

10 Refrigerator Appliances 600 5

2 Smartphone Electronics 500 15

6 Sofa Furniture 450 12

11 DiningTable Furniture 350 8

4 CoffeeTable Furniture 200 10

3 DeskChair Furniture 150 20

9 Microwave Appliances 120 8

5 Printer Electronics 100 5

13 Shoes Clothing 70 40

12 Headphones Electronics 60 25

14 Blender Appliances 50 10

8 Jeans Clothing 40 30

7 T-shirt Clothing 20 50

8.)List the products in descending order of stock quantity and within the same quantity, in alphabetical order of product name.

SELECT \* FROM Products ORDER BY StockQuantity DESC, ProductName;

OUTPUT:=

PRODUCTID PRODUCTNAME CATEGORY PRICE STOCKQUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7 T-shirt Clothing 20 50

13 Shoes Clothing 70 40

8 Jeans Clothing 40 30

12 Headphones Electronics 60 25

3 DeskChair Furniture 150 20

2 Smartphone Electronics 500 15

6 Sofa Furniture 450 12

14 Blender Appliances 50 10

4 CoffeeTable Furniture 200 10

11 DiningTable Furniture 350 8

9 Microwave Appliances 120 8

15 TV Electronics 900 6

5 Printer Electronics 100 5

10 Refrigerator Appliances 600 5

--For the Orders Table:

1.)Calculate the total amount spent by each customer and list the customers in alphabetical order by name.

SELECT CustomerName, SUM(TotalAmount) AS TotalAmountSpent FROM Orders GROUP BY CustomerName

ORDER BY CustomerName;

OUTPUT:=

CUSTOMERNAME TOTAL\_AMOUNT\_SPENT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams 750

BobJohnson 600

ChrisRoberts 350

DavidBrown 400

EmilyThomas 200

Jane Doe 450

John Smith 300

LauraGarcia 600

LindaLee 350

Mark Harris 950

Mary Davis 550

Mike Clark 900

PeterEvans 450

Sara Martin 800

TomWilson 700

2.)Find the customer who spent the most in a single order.

SELECT CustomerName, MAX(TotalAmount) AS MaxTotalAmount FROM Orders GROUP BY CustomerName ORDER BY MaxTotalAmount DESC

LIMIT 1;

OUTPUT:=

(pending)

3.)List the orders placed on or after '2023-10-23'.

SELECT \* FROM Orders WHERE OrderDate >= '2023-10-23' ORDER BY OrderDate;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATE TOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9 MikeClark 23-OCT-23 900

10 EmilyThomas 24-OCT-23 200

11 PeterEvans 25-OCT-23 450

12 SaraMartin 26-OCT-23 800

13 ChrisRoberts 27-OCT-23 350

14 LauraGarcia 28-OCT-23 600

15 MarkHarris 29-OCT-23 950

4.)Calculate the average order amount.

SELECT AVG(TotalAmount) AS AverageOrderAmount FROM Orders;

OUTPUT:=

AVERAGE\_ORDER\_AMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

556.666667

5.)List the customers who placed more than one order.

SELECT CustomerName FROM Orders GROUP BY CustomerName HAVING COUNT(OrderID) > 1;

OUTPUT:=

norowsselected

6.)Find the customer who placed the earliest order.

SELECT CustomerName, MIN(OrderDate) AS EarliestOrderDate FROM Orders GROUP BY CustomerName ORDER BY EarliestOrderDate

LIMIT 1;

OUTPUT:=

(pending)

7.)Count the number of orders placed in October 2023.

SELECT COUNT(\*) AS NumberOfOrders FROM Orders WHERE OrderDate >= '2023-10-01' AND OrderDate <= '2023-10-31';

OUTPUT:=

ORDERS\_IN\_OCTOBER\_COUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15

8.)List the orders in ascending order of total amount.

SELECT \* FROM Orders ORDER BY TotalAmount;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATE TOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10 EmilyThomas 24-OCT-23 200

1 JohnSmith 15-OCT-23 300

13 ChrisRoberts 27-OCT-23 350

8 LindaLee 22-OCT-23 350

5 DavidBrown 19-OCT-23 400

2 Jane Doe 16-OCT-23 450

11 PeterEvans 25-OCT-23 450

6 MaryDavis 20-OCT-23 550

3 BobJohnson 17-OCT-23 600

14 LauraGarcia 28-OCT-23 600

7 TomWilson 21-OCT-23 700

4 AliceWilliams 18-OCT-23 750

12 SaraMartin 26-OCT-23 800

9 MikeClark 23-OCT-23 900

15 MarkHarris 29-OCT-23 950

--For the Customers Table:

1.)List the customers in a specific city, e.g., 'New York'.

SELECT CustomerName FROM Customers WHERE City = 'New York';

OUTPUT:=

no rows selected

2.)Find the state with the highest number of customers.

SELECT State, COUNT(\*) AS CustomerCount FROM Customers GROUP BY State ORDER BY CustomerCount DESC LIMIT 1;

OUTPUT:=

STATE CUSTOMER\_COUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TX 4

CA 3

AZ 1

CO 1

MA 1

IL 1

FL 1

WA 1

PA 1

3.)List the customers in alphabetical order by name.

SELECT CustomerName FROM Customers ORDER BY CustomerName;

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams

Bob Johnson

Chris Roberts

David Brown

Emily Thomas

JaneDoeLaura

Garcia

LindaLeeMark

Harris Mary

Davis Mike

Clark Peter

Evans Sara

Martin Tom

Wilson

4.)Count the total number of customers.

SELECT COUNT(\*) AS TotalCustomers FROM Customers;

OUTPUT:=

TOTAL\_CUSTOMERS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

14

5.)Find the customer with the longest name (maximum character length).

SELECT CustomerName FROM Customers ORDER BY LENGTH(CustomerName) DESC LIMIT 1;

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams

Chris Roberts

Laura Garcia

Emily Thomas

Bob Johnson

Sara Martin

Mark Harris

David Brown

Peter Evans

Mike Clark

MaryDavisTom

Wilson

LindaLeeJane

Doe

6.)List the customers in a specific state, e.g., 'TX'.

SELECT CustomerName FROM Customers WHERE State = 'TX';

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams

Tom Wilson

Mike Clark

Emily Thomas

7.)Calculate the average customer name length.

SELECT AVG(LENGTH(CustomerName)) AS AverageNameLength FROM Customers;

OUTPUT:=

AVERAGE\_NAME\_LENGTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10.9285714

8.)Find the state with the fewest customers.

SELECT State, COUNT(\*) AS CustomerCount FROM Customers GROUP BY State ORDER BY CustomerCount LIMIT 1;

STATE CUSTOMER\_COUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WA 1

FL 1

AZ 1

CO 1

PA 1

IL 1

MA 1

CA 3

TX 4

------------------------------------------GROUP BY - HAVING CLAUSE-----------------------------------------

--For the Employees Table:=

1.)Find the average salary for departments with more than 2 employees.

SELECT Department, AVG(Salary) AS AverageSalary FROM Employees GROUP BY Department HAVING COUNT(EmployeeID) > 2;

OUTPUT:=

DEPARTMENT AVERAGE\_SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HR 52333.3333

Marketing 55000

IT 62200

Finance 58500

2.)List the departments with at least 3 employees and calculate the total salary for each.

SELECT Department, COUNT(EmployeeID) AS EmployeeCount, SUM(Salary) AS TotalSalary FROM Employees GROUP BY Department

HAVING COUNT(EmployeeID) >= 3;

OUTPUT:=

DEPARTMENT TOTAL\_SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HR 157000

Marketing 165000

IT 311000

Finance 234000

3.)Find the department with the highest average salary for employees earning more than $55,000.

SELECT Department, AVG(Salary) AS AverageSalary FROM Employees GROUP BY Department HAVING AVG(Salary) > 55000 ORDER BY AverageSalary DESC

LIMIT 1;

OUTPUT:=

DEPARTMENT AVERAGE\_SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT 62200

Finance 58500

4.)Calculate the total salary for departments where the minimum salary is less than $55,000.

SELECT Department, SUM(Salary) AS TotalSalary FROM Employees GROUP BY Department HAVING MIN(Salary) < 55000;

OUTPUT:=

DEPARTMENT TOTAL\_SALARY

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

HR 157000

Marketing 165000

5.)List the departments with an average salary above $58,000 and more than 2 employees.

SELECT Department, AVG(Salary) AS AverageSalary, COUNT(EmployeeID) AS EmployeeCount FROM Employees GROUP BY Department

HAVING AVG(Salary) > 58000 AND COUNT(EmployeeID) > 2;

OUTPUT:=

DEPARTMENT AVERAGE\_SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT 62200

Finance 58500

6.)Find the department with the highest total salary for employees with salaries between $50,000 and $60,000.

SELECT Department, SUM(Salary) AS TotalSalary

FROM Employees

WHERE Salary BETWEEN 50000 AND 60000

GROUP BY Department

HAVING TotalSalary = DESC;

OUTPUT:=

DEPARTMENT TOTALSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Finance 234000

Marketing 165000

HR 157000

IT 60000

7.)List the departments with exactly 2 employees and find the maximum salary in each.

SELECT Department, MAX(Salary) AS MaxSalary FROM Employees GROUP BY Department HAVING COUNT(EmployeeID) = 2;

OUTPUT:=

no rows selected

8.)Calculate the average salary for the Marketing department and list it only if its above $55,000.

SELECT Department, AVG(Salary) AS AverageSalary FROM Employees WHERE Department = 'Marketing' GROUP BY Department

HAVING AVG(Salary) > 55000;

OUTPUT:=

(pending)

--For the Products Table:=

1.)Calculate the average price for products with a stock quantity greater than 10.

SELECT Category, AVG(Price) AS AveragePrice FROM Products GROUP BY Category HAVING SUM(StockQuantity) > 10;

OUTPUT:=

AVERAGE\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

184.285714

2.)List the categories with at least 5 products and find the maximum price in each.

SELECT Category, MAX(Price) AS MaximumPrice FROM Products GROUP BY Category HAVING COUNT(\*) >= 5;

OUTPUT:=

CATEGORY MAX\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 900

3.)Find the category with the highest average price for products costing less than $100.

SELECT Category, AVG(Price) AS AveragePrice FROM Products GROUP BY Category HAVING AVG(Price) < 100 ORDER BY AveragePrice DESC

LIMIT 1;

OUTPUT:=

CATEGORY AVERAGE\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 60

Appliances 50

Clothing 43.3333333

4.)Calculate the total stock quantity for categories where the minimum price is above $30.

SELECT Category, MAX(Price) AS MaximumPrice FROM Products GROUP BY Category HAVING COUNT(\*) >= 5;

OUTPUT:=

CATEGORY TOTAL\_STOCK\_QUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 61

Furniture 50

Appliances 23

5.)List the categories with an average price below $70 and at least 3 products.

SELECT Category, AVG(Price) AS AveragePrice, COUNT(\*) AS ProductCount FROM Products GROUP BY Category

HAVING AVG(Price) < 70 AND COUNT(\*) >= 3;

OUTPUT:=

CATEGORY AVGPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Clothing 43.3333333

6.)Find the category with the lowest total stock quantity for products with prices above $50.

SELECT Category, SUM(StockQuantity) AS TotalStockQuantity FROM Products GROUP BY Category HAVING MIN(Price) > 50 ORDER BY

TotalStockQuantity LIMIT 1;

OUTPUT:=

CATEGORY TOTAL\_STOCK\_QUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Appliances 13

Clothing 40

Furniture 50

Electronics 51

7.)List the categories with exactly 4 products and find the minimum price in each.

SELECT Category, MIN(Price) AS MinimumPrice FROM Products GROUP BY Category HAVING COUNT(\*) = 4;

OUTPUT:=

CATEGORY MIN\_PRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Furniture 150

8.)Calculate the average price for the Electronics category and list it only if its below $75.

SELECT Category, AVG(Price) AS AveragePrice FROM Products WHERE Category = 'Electronics' GROUP BY Category

HAVING AVG(Price) < 75;

OUTPUT:=

(pending)

--For the Orders Table:=

1.)Calculate the total amount spent by customers who placed more than 2 orders.

SELECT CustomerName, SUM(TotalAmount) AS TotalAmountSpent FROM Orders GROUP BY CustomerName HAVING COUNT(OrderID) > 2;

OUTPUT:=

no rows selected

2.)List the customers who spent at least $500 in a single order and find the highest total amount spent by each.

SELECT CustomerName, MAX(TotalAmount) AS HighestTotalAmount FROM Orders GROUP BY CustomerName HAVING MAX(TotalAmount) >= 500;

OUTPUT:=

CUSTOMERNAME HIGHESTTOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mary Davis 550

Mark Harris 950

TomWilson 700

Sara Martin 800

AliceWilliams 750

LauraGarcia 600

BobJohnson 600

Mike Clark 900

3.)Find the orders placed on or after '2023-10-23' with a total amount greater than $400.

SELECT OrderID, OrderDate, TotalAmount FROM Orders WHERE OrderDate >= '2023-10-23' GROUP BY OrderID, OrderDate, TotalAmount

HAVING TotalAmount > 400;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATE TOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9 MikeClark 23-OCT-23 900

11 PeterEvans 25-OCT-23 450

12 SaraMartin 26-OCT-23 800

14 LauraGarcia 28-OCT-23 600

15 MarkHarris 29-OCT-23 950

4.)Calculate the average order amount for orders with more than 1 item.

SELECT AVG(TotalAmount) AS AverageOrderAmount FROM Orders GROUP BY OrderID HAVING COUNT(\*) > 1;

OUTPUT:=

no rows selected

5.)List the customers who placed exactly 1 order and find the minimum total amount spent.

SELECT CustomerName, MIN(TotalAmount) AS MinimumTotalAmount FROM Orders GROUP BY CustomerName HAVING COUNT(\*) = 1;

OUTPUT:=

CUSTOMERNAME MINIMUM\_AMOUNT\_SPENT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DavidBrown 400

Mary Davis 550

Mark Harris 950

TomWilson 700

LindaLee 350

PeterEvans 450

Sara Martin 800

AliceWilliams 750

EmilyThomas 200

Jane Doe 450

LauraGarcia 600

John Smith 300

BobJohnson 600

Mike Clark 900

ChrisRoberts 350

6.)Find the orders placed in October 2023 with a total amount less than $600.

SELECT OrderID, OrderDate, TotalAmount FROM Orders WHERE DATE\_PART('year', OrderDate) = 2023 AND DATE\_PART('month', OrderDate) = 10

GROUP BY OrderID, OrderDate, TotalAmount HAVING TotalAmount < 600;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATETOT ALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 JohnSmith 15-OCT-23 300

2 JaneDoe 16-OCT-23 450

5 DavidBrown 19-OCT-23 400

6 MaryDavis 20-OCT-23 550

8 LindaLee 22-OCT-23 350

10 EmilyThomas 24-OCT-23 200

11 PeterEvans 25-OCT-23 450

13 ChrisRoberts 27-OCT-23 350

7.)Calculate the total amount spent by customers with names longer than 10 characters.

SELECT CustomerName, SUM(TotalAmount) AS TotalAmountSpent FROM Orders

GROUP BY CustomerName HAVING LENGTH(CustomerName) > 10;

OUTPUT:=

CUSTOMERNAME TOTAL\_AMOUNT\_SPENT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DavidBrown 400

MarkHarris 950

PeterEvans 450

Sara Martin 800

AliceWilliams 750

EmilyThomas 200

LauraGarcia 600

BobJohnson 600

ChrisRoberts 350

8.)List the orders with at least 2 items and find the maximum total amount.

SELECT OrderID, MAX(TotalAmount) AS MaximumTotalAmount FROM Orders GROUP BY OrderID HAVING COUNT(\*) >= 2;

OUTPUT:=

no rows selected

--For the Customers Table:=

1.)List the customers in a specific city, e.g., 'New York', and find the total number of customers in that city.

SELECT City, COUNT(\*) AS CustomerCount FROM Customers WHERE City = 'New York' GROUP BY City;

OUTPUT:=

no rows selected

2.)Find the states with more than 2 customers and calculate the average name length for customers in each state.

SELECT State, AVG(LENGTH(CustomerName)) AS AverageNameLength FROM Customers GROUP BY State HAVING COUNT(\*) > 2;

OUTPUT:=

STATE AVG\_NAME\_LENGTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TX 11.5

CA 9.33333333

3.)List the customers in alphabetical order by name, but only if their names contain the letter 'a'.

SELECT CustomerName FROM Customers GROUP BY CustomerName HAVING CustomerName LIKE '%a%'

ORDER BY CustomerName;

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams

David Brown

Emily Thomas

JaneDoeLaura

Garcia

LindaLeeMark

Harris Mary

Davis Mike

Clark Peter

Evans Sara

Martin

4.)Count the total number of customers in each state and find the states with exactly 1 customer.

SELECT State, COUNT(\*) AS CustomerCount FROM Customers GROUP BY State HAVING CustomerCount = 1;

OUTPUT:=

STATE TOTALCUSTOMERS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WA 1

FL 1

AZ 1

CO 1

PA 1

MA 1

IL 1

5.)Find the customer with the longest name (maximum character length) and their state.

SELECT CustomerName, State FROM Customers GROUP BY CustomerName, State HAVING LENGTH(CustomerName) = (SELECT MAX(LENGTH(CustomerName)) FROM Customers);

OUTPUT:=

CUSTOMERNAME STATE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams TX

ChrisRoberts CO

LauraGarcia MA

EmilyThomas TX

BobJohnson IL

Sara Martin WA

Mark Harris FL

DavidBrown PA

PeterEvans CA

Mike Clark TX

Mary Davis AZ

TomWilson TX

LindaLee CA

Jane Doe CA

6.)List the customers in a specific state, e.g., 'TX', and calculate the total number of customers in that state.

SELECT State, COUNT(\*) AS CustomerCount FROM Customers WHERE State = 'TX' GROUP BY State;

OUTPUT:=

CUSTOMERNAME TOTAL\_CUSTOMERS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TomWilson 1

AliceWilliams 1

EmilyThomas 1

Mike Clark 1

7.)Calculate the average customer name length for customers in states with more than 3 customers.

SELECT State, AVG(LENGTH(CustomerName)) AS AverageNameLength FROM Customers GROUP BY State HAVING COUNT(\*) > 3;

OUTPUT:=

STATE AVG\_NAME\_LENGTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TX 11.5

8.)Find the states with the fewest customers and list the customers in those states.

SELECT State, CustomerName

FROM Customers

WHERE State IN (

SELECT State

FROM Customers

GROUP BY State

HAVING COUNT(\*) = (

SELECT MIN(CustomerCount)

FROM (

SELECT COUNT(\*) AS CustomerCount

FROM Customers

GROUP BY State

) AS Subquery

)

);

OUTPUT:=

(pending)

------------------------------------------------GROUPBY-HAVING AND ORDERBY------------------------------------

--For the Employees Table:=

1.)List the departments with at least 2 employees,calculatethetotalsalary for each, and order them by total salary in

descending order.

SELECT Department,SUM(Salary)ASTotalSalary FROM Employees GROUP BY Department HAVING COUNT(\*) >= 2 ORDERBY TotalSalary DESC;

OUTPUT:=

DEPARTMENT TOTALSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT 311000

Finance 234000

Marketing 165000

HR 157000

2.)Find the department with the highest average salary for employees earning morethan $55,000,andordertheresults

by the averagesalary indescendingorder.

SELECT Department,AVG(Salary)ASAvgSalary FROM Employees WHERE Salary > 55000 GROUP BY Department ORDERBY AvgSalaryDESC;

OUTPUT:=

DEPARTMENT AVGSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT 62200

Finance 58500

Marketing 56000

3.)List the departments and the maximum salary in each department, and order them by the maximum salary in ascending order.

SELECT Department,MAX(Salary)ASMaxSalary FROM Employees GROUP BY Department

ORDER BY MaxSalaryASC;

OUTPUT:=

DEPARTMENT MAXSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HR 55000

Marketing 56000

Finance 60000

IT 65000

4.)Find the department with the lowest average salary for employees with salaries less than $60,000 and order the results by the average salary in ascending order.

SELECT Department, AVG(Salary) AS AvgSalary FROM Employees WHERE Salary < 60000 GROUP BY Department ORDERBY AvgSalaryASC

LIMIT 1;

OUTPUT:=

(pending)

5.)Calculate the total salary for departments where the minimum salary is less than $55,000, and order the results by total salary in descending order.

SELECT Department,SUM(Salary)ASTotalSalary FROM Employees GROUP BY Department HAVING MIN (Salary)<55000

ORDER BY TotalSalaryDESC;

OUTPUT:=

DEPARTMENT TOTALSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Marketing 165000

HR 157000

6.)List the departments with more than 3 employees, find the average salary in each department, and order them by the average salary in descending order.

SELECT Department,AVG(Salary)ASAvgSalary FROM Employees

GROUP BY Department

HAVING COUNT(\*)>3

ORDER BY AvgSalaryDESC;

OUTPUT:=

DEPARTMENT AVGSALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT 62200

Finance 58500

7.)Find the departments with exactly 2 employees, list the employees in each department, and order the results by department and then by employee last name in ascending order.

SELECTE1.Department,E1.FirstName,E2.FirstName

FROM Employees E1

INNERJOINEmployeesE2ON E1.Department=E2.DepartmentANDE1.EmployeeID<E2.EmployeeID GROUP BY

E1.Department, E1.FirstName, E2.FirstName

HAVINGCOUNT(\*)=2

ORDERBYE1.Department,E1.FirstName,E2.FirstName;

OUTPUT:=

no rows selected

8.)List the employees in the IT department in alphabetical order by last name, find the average salary in that department, and order the results by average salary in ascending order.

SELECT FirstName,Salary

FROM Employees

WHERE Department='IT' ORDER

BY FirstName;

OUTPUT:-

FIRSTNAME SALARY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Alice 65000

Bob 60000

Chris 61000

David 62000

Emily 63000

--For the Products Table:+

1.)List the categories with at least 5 products, find the minimum price in each category, and order the results by category in ascending order.

SELECTCategory,MIN(Price)ASMinPrice

FROM Products

GROUP BY Category

HAVINGCOUNT(\*)>=5

ORDERBYCategoryASC;

OUTPUT:=

no rows selected

2.)Find the category with the highest average price for products costing less than $100 and order the results by the average price in descending order.

SELECTCategory,AVG(Price)ASAvgPrice

FROM Products

WHEREPrice<100

GROUPBYCategoryORDERBYA

vgPriceDESC;

OUTPUT:=

CATEGORY AVGPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 60

Appliances 50

Clothing 43.3333333

3.)Calculate the total stock quantity for categories where the minimum price is above $30 and order the results by total stock quantity in descending order.

SELECTCategory,SUM(StockQuantity)ASTotalStockQuantity FROM

Products

GROUP BY Category

HAVINGMIN(Price)>30

ORDERBYTotalStockQuantityDESC;

OUTPUT:=

CATEGORY TOTALSTOCKQUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 51

Furniture 50

Appliances 23

4.)List the categories and the maximum price in each category, find the category with the lowest maximum price, and order the results by category in ascending order.

SELECTCategory,MAX(Price)ASMaxPrice

FROM Products

GROUPBYCategory ORDERBYMaxPrice;

OUTPUT:=

CATEGORY MAXPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Clothing 70

Furniture 450

Appliances 600

Electronics 900

5.)Find the categories with an average price below $70 and at least 3 products, and order the results by the average price in ascending order.

SELECTCategory,AVG(Price)ASAvgPrice

FROM Products

GROUPBYCategory

HAVINGAVG(Price)<70ANDCOUNT(\*)>=3

ORDERBYAvgPriceASC;

OUTPUT:=

CATEGORY AVGPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Clothing 43.3333333

6,)Calculate the average price for products with a stock quantity greater than 10, and order the results by average price in descending order.

SELECTCategory,AVG(Price)ASAvgPrice

FROM Products

WHEREStockQuantity>10

GROUP BY Category

ORDERBYAvgPriceDESC;

OUTPUT:=

CATEGORY AVGPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Furniture 300

Electronics 280

Clothing 43.3333333

7.)List the categories with exactly 4 products, find the maximum price in each category, and order the results by category in descending order.

SELECTCategory,MAX(Price)ASMaxPrice

FROM Products

GROUP BY Category

HAVINGCOUNT(\*)=4

ORDERBYCategoryDESC;

OUTPUT:=

CATEGORY MAXPRICE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Furniture 450

Electronics 900

8.)Find the category with the highest total stock quantity for products with prices above $50 and order the results by the total stock quantity in descending order.

SELECTCategory,SUM(StockQuantity)ASTotalStockQuantity FROM

Products

WHERE Price >50

GROUPBYCategory

ORDERBYTotalStockQuantityDESC;

OUTPUT:=

CATEGORY TOTALSTOCKQUANTITY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Electronics 51

Furniture 50

Clothing 40

Appliances 13

--For the Orders Table:==

1.)Calculate the total amount spent by customers who placed more than 2 orders, list the customers in alphabetical order, and order the results by total amount in descending order.

SELECTCustomerName,SUM(TotalAmount)ASTotalAmount FROM

Orders

GROUPBYCustomerName HAVING

COUNT(\*) >2

ORDERBYCustomerNameASC,TotalAmountDESC;

OUTPUT:=

no rows selected

2.)List the customers who spent at least $500 in a single order, find the highest total amount spent by each customer, and order the results by customer name in ascending order.

SELECTCustomerName,MAX(TotalAmount)ASHighestAmount FROM

Orders

GROUPBYCustomerName

HAVINGMAX(TotalAmount)>=500 ORDER

BY CustomerName ASC;

OUTPUT:=

CUSTOMERNAME HIGHESTAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams 750

BobJohnson 600

LauraGarcia 600

Mark Harris 950

Mary Davis 550

Mike Clark 900

Sara Martin 800

TomWilson 700

3.)Find the orders placed on or after '2023-10-23' with a total amount greater than $400 and order the results by order date in ascending order.

SELECT\* FROM Orders WHEREOrderDate>=to\_date('2023-10-23','YYYY=MM-DD')ANDTotalAmount>400 ORDER BY

OrderDate ASC;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATE TOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9 MikeClark 23-OCT-23 900

11 PeterEvans 25-OCT-23 450

12 SaraMartin 26-OCT-23 800

14 LauraGarcia 28-OCT-23 600

15 MarkHarris 29-OCT-23 950

4.)Calculate the average order amount for orders with more than 1 item, list the orders in descending order by order amount, and within the same amount, order them by order date in ascending order.

SELECTOrderID,AVG(TotalAmount)ASAverageAmount

FROM Orders

GROUP BY OrderID

HAVINGCOUNT(\*)>1

ORDERBYAverageAmount DESC,OrderDate;

OUTPUT:=

(pending.)

5.)List the customers who placed exactly 1 order, find the minimum total amount spent, and order the results by customer name in ascending order.

SELECTCustomerName,MIN(TotalAmount)ASMinAmount FROM

Orders

GROUPBYCustomerName HAVING

COUNT(\*) = 1

ORDERBYCustomerNameASC;

OUTPUT:=

CUSTOMERNAME MINAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams 750

BobJohnson 600

ChrisRoberts 350

DavidBrown 400

EmilyThomas 200

Jane Doe 450

John Smith 300

LauraGarcia 600

LindaLee 350

Mark Harris 950

Mary Davis 550

Mike Clark 900

PeterEvans 450

Sara Martin 800

TomWilson 700

6.)Find the orders placed in October 2023 with a total amount less than $600, list the orders in descending order by order date, and within the same date, order them by total amount in ascending order.

SELECT\* FROMOrders

WHEREOrderDate>=to\_date('2023-10-01','YYYY-MM-DD')ANDOrderDate<to\_date('2023-11-

01','YYYY\_MM\_DD') AND TotalAmount <600

ORDERBY OrderDateDESC,TotalAmountASC;

OUTPUT:=

ORDERID CUSTOMERNAME ORDERDATE TOTALAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

13 ChrisRoberts 27-OCT-23 350

11 PeterEvans 25-OCT-23 450

10 EmilyThomas 24-OCT-23 200

8 LindaLee 22-OCT-23 350

6 MaryDavis 20-OCT-23 550

5 DavidBrown 19-OCT-23 400

2 Jane Doe 16-OCT-23 450

1 JohnSmith 15-OCT-23 300

7.)Calculate the total amount spent by customers with names longer than 10 characters, list the customers in ascending order by total amount, and within the same amount, order them by customer name in descending order.

SELECTCustomerName,SUM(TotalAmount)ASOrderAmount FROM

Orders

GROUPBYCustomerName

HAVINGLENGTH(CustomerName)>10

ORDERBYOrderAmount,CustomerNameDESC;

OUTPUT:=

CUSTOMERNAME ORDERAMOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EmilyThomas 200

ChrisRoberts 350

DavidBrown 400

PeterEvans 450

LauraGarcia 600

BobJohnson 600

AliceWilliams 750

Sara Martin 800

Mark Harris 950

8.)List the orders with at least 2 items, find the maximum total amount, and order the results by maximum amount in descending order.

SELECTOrderID,MAX(TotalAmount)ASMaxAmount FROM

Orders

GROUP BY OrderID

HAVINGCOUNT(\*)>=2

ORDERBYMaxAmountDESC;

OUTPUT:=

no rows selected

--For the Customers Table:=

1.)List the customers in a specific city, e.g., 'New York', and find the total number of customers in that city, order the results by customer name in ascending order.

SELECTCustomerName

FROM Customers

WHERE City = 'New

York'ORDERBYCustomerNameAS

C;

OUTPUT:=

no rows selected

2.)Find the states with more than 2 customers, calculate the average name length for customers in each state, and order the results by state in ascending order.

SELECTState,AVG(LENGTH(CustomerName))ASAvgNameLength FROM

Customers

GROUP BY State

HAVINGCOUNT(\*)>2

ORDERBYState ASC;

OUTPUT:=

STATE AVGNAMELENGTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CA 9.33333333

TX 11.5

3.)List the customers in alphabetical order by name, but only if their names contain the letter 'a', and order the results by customer name in ascending order.

SELECTCustomerName

FROM Customers

WHERECustomerNameLIKE'%a%'ORD

ER BY CustomerName ASC;

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams

David Brown

Emily Thomas

JaneDoeLaura

Garcia

LindaLeeMark

Harris Mary

Davis Mike

Clark Peter

Evans Sara

Martin

4.)Count the total number of customers in each state and find the states with exactly 1 customer, order the results by state in descending order.

SELECT State,COUNT(\*)AS CustomerCount FROM Customers GROUP BY State

HAVING COUNT(\*) = 1 ORDER BY StateDESC;

OUTPUT:=

STATE CUSTOMERCOUNT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WA 1

PA 1

MA 1

IL 1

FL 1

CO 1

AZ 1

5.)Find the customer with the longest name (maximum character length) and their state.

SELECT CustomerName,State FROM Customers ORDER BY LENGTH(CustomerName)DESC;

OUTPUT:=

CUSTOMERNAME STATE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AliceWilliams TX

ChrisRoberts CO

LauraGarcia MA

EmilyThomas TX

BobJohnson IL

Sara Martin WA

Mark Harris FL

DavidBrown PA

PeterEvans CA

Mike Clark TX

Mary Davis AZ

TomWilson TX

LindaLee CA

Jane Doe CA

6.)List the customers in a specific state, e.g., 'TX', and calculate the total number of customers in that state, order the results by customer name in descending order.

SELECT CustomerName FROM Customers WHERE State = 'TX' ORDER BY CustomerNameDESC;

OUTPUT:=

CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tom Wilson

Mike Clark

Emily Thomas

AliceWilliams

7.)Calculate the average customer name length for customers in states with more than 3 customers and order the results by average name length in descending order.

SELECT State,AVG(LENGTH(CustomerName))AS AvgName Length FROM Customers GROUP BY State HAVING COUNT(\*)>3 ORDER BY AvgName Length DESC;

OUTPUT:=

STATE AVGNAMELENGTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TX 11.5

8.)Find the states with the fewest customers, list the customers in those states, and order the results by state in ascending order.

SELECT State,CustomerName FROM Customers WHERE StateIN(SELECT State FROM Customers GROUP BY StateHAVING COUNT(\*)=1) ORDER BY

State ASC;

OUTPUT:=

STATE CUSTOMERNAME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AZ Mary Davis

CO ChrisRoberts

FL MarkHarris

IL BobJohnson

MA LauraGarcia

PA DavidBrown

WA SaraMartin