Superset Id:-6361766

**Advanced SQL Concepts**

**Exercise 1: Ranking and Window Functions**

**Cus.sql**

**CREATE TABLE Products (**

**ProductID INT PRIMARY KEY,**

**ProductName VARCHAR(100),**

**Category VARCHAR(50),**

**Price DECIMAL(10, 2)**

**);**

**CREATE TABLE Customers (**

**CustomerID INT PRIMARY KEY,**

**Name VARCHAR(100),**

**Region VARCHAR(50)**

**);**

**CREATE TABLE Orders (**

**OrderID INT PRIMARY KEY,**

**CustomerID INT,**

**OrderDate DATE,**

**FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)**

**);**

**CREATE TABLE OrderDetails (**

**OrderDetailID INT PRIMARY KEY,**

**OrderID INT,**

**ProductID INT,**

**Quantity INT,**

**FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),**

**FOREIGN KEY (ProductID) REFERENCES Products(ProductID)**

**);**

**-- Sample Data**

**INSERT INTO Customers (CustomerID, Name, Region) VALUES**

**(1, 'Alice', 'North'),**

**(2, 'Bob', 'South'),**

**(3, 'Charlie', 'East'),**

**(4, 'David', 'West');**

**INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES**

**(1, 'Laptop', 'Electronics', 1200.00),**

**(2, 'Smartphone', 'Electronics', 800.00),**

**(3, 'Tablet', 'Electronics', 600.00),**

**(4, 'Headphones', 'Accessories', 150.00);**

**INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES**

**(1, 1, '2023-01-15'),**

**(2, 2, '2023-02-20'),**

**(3, 3, '2023-03-25'),**

**(4, 4, '2023-04-30');**

**INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES**

**(1, 1, 1, 1),**

**(2, 2, 2, 2),**

**(3, 3, 3, 1),**

**(4, 4, 4, 3);**

**SELECT \***

**FROM (**

**SELECT**

**ProductID,**

**ProductName,**

**Category,**

**Price,**

**ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum**

**FROM Products**

**) AS RankedProducts**

**WHERE RowNum <= 3;**

**SELECT \***

**FROM (**

**SELECT**

**ProductID,**

**ProductName,**

**Category,**

**Price,**

**RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum**

**FROM Products**

**) AS RankedProducts**

**WHERE RankNum <= 3;**

**SELECT \***

**FROM (**

**SELECT**

**ProductID,**

**ProductName,**

**Category,**

**Price,**

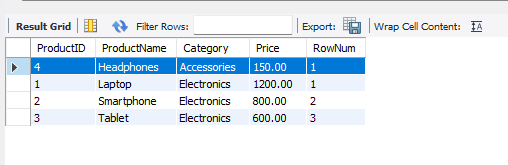
**DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankNum**

**FROM Products**

**) AS RankedProducts**

**WHERE DenseRankNum <= 3;**

**Output:-**

****

**Exercise 1: Create a Stored Procedure**

. Define the stored procedure with a parameter for DepartmentID

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

SELECT

EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

1. Write the SQL query to select employee details based on the DepartmentID.

-> EXEC sp\_GetEmployeesByDepartment @DepartmentID = 3;

Output:-

| **EmployeeID** | **FirstName** | **LastName** | **DepartmentID** | **Salary** | **JoinDate** |
| --- | --- | --- | --- | --- | --- |
| 3 | Michael | Johnson | 3 | 7000.00 | 2018-07-30 |

3.3. Create a stored procedure named `sp\_InsertEmployee` with the following code: CREATE PROCEDURE sp\_InsertEmployee @FirstName VARCHAR(50), @LastName VARCHAR(50), @DepartmentID INT, @Salary DECIMAL(10,2), @JoinDate DATE AS BEGIN INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate) VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate); END;

CREATE PROCEDURE sp\_InsertEmployee

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

AS

BEGIN

INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);

END;

Execute

EXEC sp\_InsertEmployee

@FirstName = 'Sarah',

@LastName = 'Lee',

@DepartmentID = 2,

@Salary = 6200.00,

@JoinDate = '2023-08-01';

Ouput:-

| **EmployeeID** | **FirstName** | **LastName** | **DepartmentID** | **Salary** | **JoinDate** |
| --- | --- | --- | --- | --- | --- |
| 5 (or next ID) | Sarah | Lee | 2 | 6200.00 | 2023-08-01 |

**Exercise 5: Return Data from a Stored Procedure**

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

**Execute:-**

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 2;

Ouput:-

For DepartmentID = 2 (Finance):