**WEEK 2 MANDATORY HANDS-ON**

**Superset ID: 6365267**

**MODULE 3: SQL**

1. **Advanced SQL concepts:**

Exercise 1: ranking and window function

1. Using row\_number()

SELECT \*

FROM (

    SELECT

        ProductID,

        ProductName,

        Category,

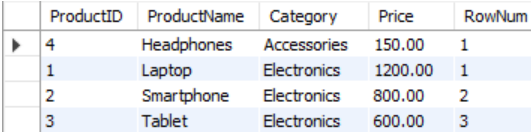
        Price,

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

    FROM Products

) AS Ranked

WHERE RowNum <= 3;



1. Using rank()

SELECT \*

FROM (

    SELECT

        ProductID,

        ProductName,

        Category,

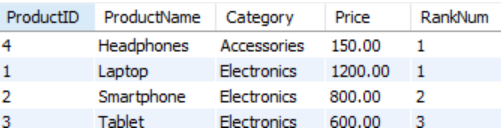
        Price,

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

    FROM Products

) AS Ranked

WHERE RankNum <= 3;



1. Using dense\_rank()

SELECT \*

FROM (

    SELECT

        ProductID,

        ProductName,

        Category,

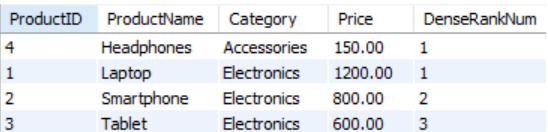
        Price,

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

    FROM Products

) AS Ranked

WHERE DenseRankNum <= 3;



1. **Stored procedure:**

Exercise 1: Create a stored procedure:

Q1. Define the stored procedure with a parameter for DepartmentID & Write the SQL query to select employee details based on the DepartmentID.

CREATE PROCEDURE sp\_GetEmployeesByDepartment

    @DeptID INT

AS

BEGIN

    SELECT EmployeeID, FirstName, LastName, Salary, JoinDate

    FROM Employees

    WHERE DepartmentID = @DeptID;

END;

EXEC sp\_GetEmployeesByDepartment @DeptID = 3;



Q2. 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;

EXEC sp\_InsertEmployee

    @FirstName = 'Alex',

    @LastName = 'Brown',

    @DepartmentID = 2,

    @Salary = 6200.00,

    @JoinDate = '2022-04-10';

EXERCISE-5: Return Data from a Stored Procedure

Q1. Define the stored procedure with a parameter for DepartmentID &  Write the SQL query to count the number of employees in the specified department.

CREATE PROCEDURE sp\_CountEmployeesByDepartment

    @DepartmentID INT

AS

BEGIN

    SELECT COUNT(\*) AS TotalEmployees

    FROM Employees

    WHERE DepartmentID = @DepartmentID;

END;

Q2. Save the stored procedure by executing the Stored procedure content

EXEC sp\_CountEmployeesByDepartment @DepartmentID = 2;

