WEEK-2 ASSIGNMENTS

SKILL:-SQL-ADVANCED CONCEPTS

**EXERCISE 1:-RANKING AND WINDOW FUNCTIONS:-**

SQL CODE:-

create database cognizant;

use cognizant;

CREATE TABLE Products (

ProductID INT,

ProductName VARCHAR(100),

Category VARCHAR(100),

Price DECIMAL(10,2)

);

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products;

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;

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

(1, 'Laptop Pro', 'Electronics', 1500.00),

(2, 'Smartphone X', 'Electronics', 1200.00),

(3, 'Smartwatch', 'Electronics', 500.00),

(4, 'Bluetooth Speaker', 'Electronics', 300.00),

(5, 'LED TV', 'Electronics', 1200.00),

(6, 'Treadmill', 'Fitness', 1000.00),

(7, 'Dumbbells', 'Fitness', 200.00),

(8, 'Exercise Bike', 'Fitness', 700.00),

(9, 'Yoga Mat', 'Fitness', 100.00),

(10, 'Resistance Bands', 'Fitness', 100.00),

(11, 'Formal Shirt', 'Clothing', 70.00),

(12, 'Leather Jacket', 'Clothing', 250.00),

(13, 'Jeans', 'Clothing', 100.00),

(14, 'Sneakers', 'Clothing', 120.00),

(15, 'T-shirt', 'Clothing', 50.00);

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products

) AS ranked\_products

WHERE rnk <= 3;

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

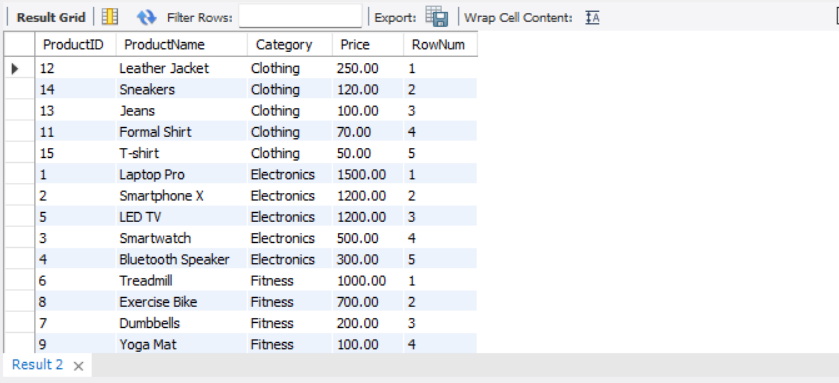
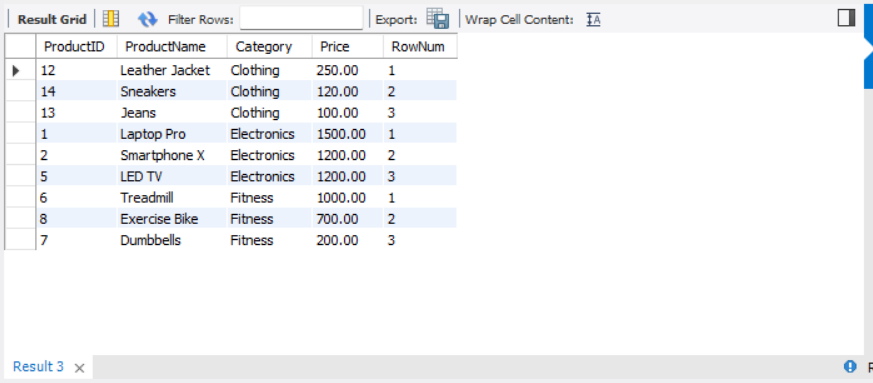
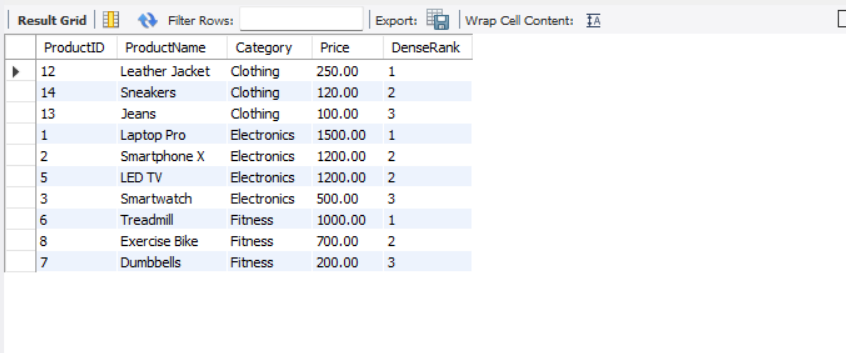
Price,

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

FROM Products

) AS Ranked

WHERE DenseRank <= 3;

OUTPUT:-

SKILL:-SQL-STORED PROCEDURE

**Exercise 1: Create a Stored Procedure**

SQL CODE:-

CREATE TABLE Employees (

EmployeeID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT,

Salary DECIMAL(10,2),

JoinDate DATE

);

DELIMITER //

CREATE PROCEDURE sp\_GetEmployeesByDepartment (

IN deptId INT

)

BEGIN

SELECT \* FROM Employees

WHERE DepartmentID = deptId;

END //

DELIMITER ;

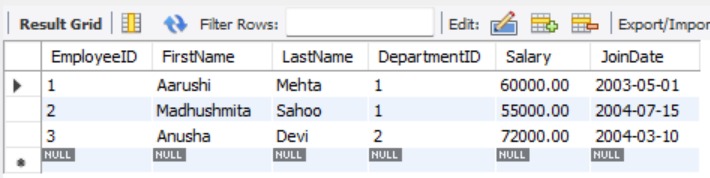
INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate) VALUES

('Arushi', 'Mehta', 1, 60000.00, '2022-05-01'),

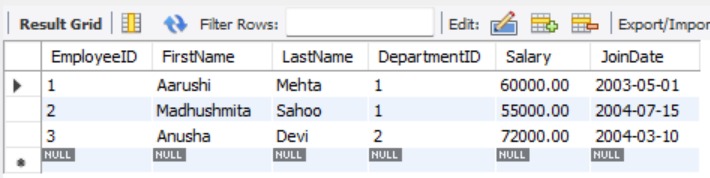
('Madhusmita', 'Sahoo', 1, 55000.00, '2021-07-15'),

('Anusha', 'Devi', 2, 72000.00, '2020-03-10');

SELECT \* FROM Employees;



CALL sp\_GetEmployeesByDepartment(1);



**EXERCISE 5: Return Data From a Stored Procedure(Count of Employees in a Department)**

SQL CODE:-

DELIMITER //

CREATE PROCEDURE sp\_CountEmployeesByDepartment (

IN deptId INT

)

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = deptId;

END //

DELIMITER ;

CALL sp\_CountEmployeesByDepartment(1);



CALL sp\_CountEmployeesByDepartment(2);

