**Cognizant Deep Skilling Week-2**

1. SQL Exercise-Advanced Concepts:

Exercise 1: Ranking and Window Functions:-

Code :-

**1.Create a New Database :**

CREATE DATABASE RankingDemo;

GO

USE RankingDemo;

GO

Step 2: Create the Products Table:

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

Step 3: Insert Sample Data into Products Table:

INSERT INTO Products VALUES

(1, 'iPhone 14', 'Electronics', 1000),

(2, 'Samsung Galaxy S23', 'Electronics', 950),

(3, 'Dell Laptop', 'Electronics', 1200),

(4, 'Blender', 'Appliances', 150),

(5, 'Microwave', 'Appliances', 300),

(6, 'Toaster', 'Appliances', 120),

(7, 'Sony Headphones', 'Electronics', 950),

(8, 'Air Conditioner', 'Appliances', 1200);

Find the top 3 most expensive products in each category using different ranking functions.

SELECT

ProductID,

ProductName,

Category,

Price,

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

RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS Rank,

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

FROM Products;

WITH RankedProducts AS (

SELECT \*,

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

FROM Products

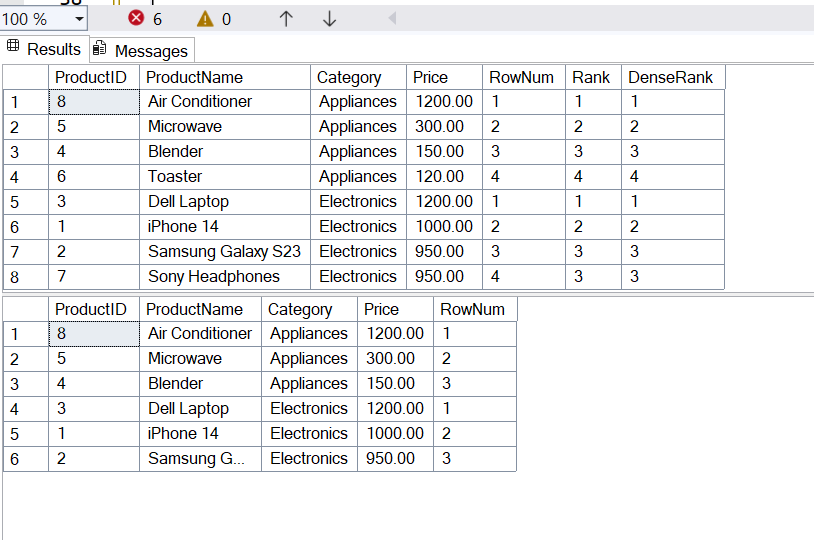
)

SELECT \*

FROM RankedProducts

WHERE RowNum <= 3;

OUTPUT:-



2.SQL Exercise-Stored Concepts:

Exercise 1: Create a Stored Procedure:-

Code:-

Step 1: Create Tables:

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

Step 2: Insert Sample Data:

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

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

('John', 'Doe', 1, 5000.00, '2020-01-15'),

('Jane', 'Smith', 2, 6000.00, '2019-03-22'),

('Michael', 'Johnson', 3, 7000.00, '2018-07-30'),

('Emily', 'Davis', 4, 5500.00, '2021-11-05');

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DeptID INT

AS

BEGIN

SELECT

EmployeeID,

FirstName,

LastName,

Salary,

JoinDate

FROM Employees

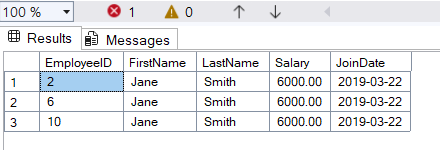
WHERE DepartmentID = @DeptID;

END;

TEST CASE:

EXEC sp\_GetEmployeesByDepartment @DeptID = 2;

OUTPUT:-



Insert New Employee:

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 = 'Alice',

@LastName = 'Brown',

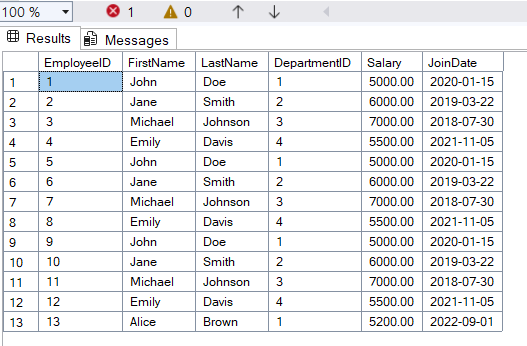
@DepartmentID = 1,

@Salary = 5200.00,

@JoinDate = '2022-09-01';

SELECT \* FROM Employees;

OUTPUT:-



Exercise 5: Return Data from a Stored Procedure:-

Code:-

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DeptID INT

AS

BEGIN

SELECT

COUNT(\*) AS EmployeeCount

FROM Employees

WHERE DepartmentID = @DeptID;

END;

EXEC sp\_GetEmployeeCountByDepartment @DeptID = 1;

OUTPUT:-

