**Week 2- Advanced SQL**

**1. SQL Exercise - Advanced concepts**

**Exercise 1: Ranking and Window Functions**

**Code:**  
CREATE DATABASE Ranking;

GO

USE Ranking;

GO

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price INT

);

GO

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

(1, 'iPhone', 'Mobile', 90000),

(2, 'Samsung', 'Mobile', 85000),

(3, 'Nokia', 'Mobile', 85000),

(4, 'Hp', 'Laptop', 78000),

(5, 'MacBook', 'Laptop', 200000);

GO

SELECT ProductID, ProductName, Category, Price,

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

FROM Products;

GO

SELECT ProductID, ProductName, Category, Price,

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

FROM Products;

GO

SELECT ProductName, Category, Price,

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

FROM Products;

GO

SELECT \* FROM (

SELECT ProductName, Category, Price,

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

FROM Products

) AS RankedProducts

WHERE RowNum <= 3;

GO

SELECT \* FROM (

SELECT ProductName, Category, Price,

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

FROM Products

) AS Ranked

WHERE rnk <= 3;

GO

SELECT \* FROM (

SELECT ProductName, Category, Price,

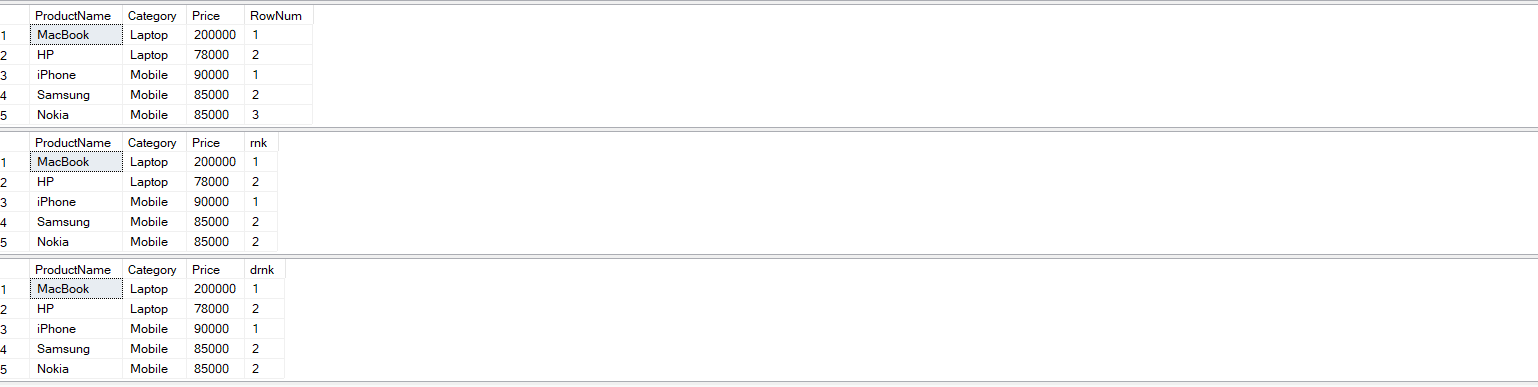
DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS drnk

FROM Products

) AS Ranked

WHERE drnk <= 3;

GO  
  
**Output:**



**4. SQL Exercise - Stored procedure**

**Exercise 1: Create a Stored Procedure**

**Code:**

CREATE DATABASE Employee;

GO

USE Employee;

GO

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

GO

CREATE TABLE Employees (

EmployeeID INT IDENTITY(1,1) PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT,

Salary DECIMAL(10,2),

JoinDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

GO

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

GO

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');

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@deptId INT

AS

BEGIN

SELECT

e.EmployeeID,

e.FirstName,

e.LastName,

d.DepartmentName,

e.Salary,

e.JoinDate

FROM Employees e

INNER JOIN Departments d ON e.DepartmentID = d.DepartmentID

WHERE e.DepartmentID = @deptId;

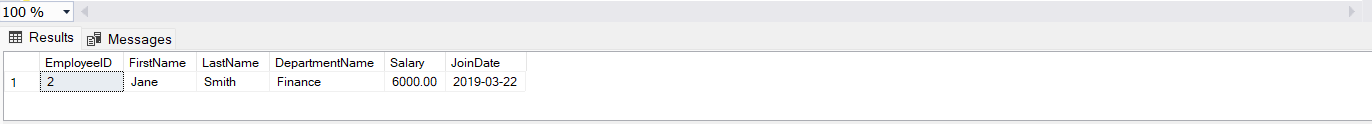
END;

GO

EXEC sp\_GetEmployeesByDepartment @deptId = 2;

GO

**Output:**

****

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;

GO

EXEC sp\_InsertEmployee

@FirstName = 'Varsha',

@LastName = 'Velavan',

@DepartmentID = 3,

@Salary = 8000.00,

@JoinDate = '2025-06-16';

GO

SELECT \* FROM Employees;

GO

**Output:**



**4. SQL Exercise - Stored procedure**

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

CREATE PROCEDURE sp\_CountEmployeesByDepartment

@deptId INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @deptId;

END;

GO

EXEC sp\_CountEmployeesByDepartment @deptId = 3;

GO  
  
**Output:**

