**1: SQL Exercise:Advanced Concepts**

***Exercise 1: Ranking and Window Functions***

CREATE TABLE Products(

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(100),

Price DECIMAL(10,2));

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

(1, 'Phone A', 'Phone', 500),

(2, 'Phone B', 'Phone', 700),

(3, 'Phone C', 'Phone', 700),

(4, 'Phone D', 'Phone', 600),

(5, 'Laptop E', 'Laptop', 1000),

(6, 'Laptop F', 'Laptop', 1200),

(7, 'Laptop G', 'Laptop', 1200),

(8, 'Laptop H', 'Laptop', 1100);

**all products along with their ranking within each category based on descending price:**

SELECT

ProductID,

ProductName,

Category,

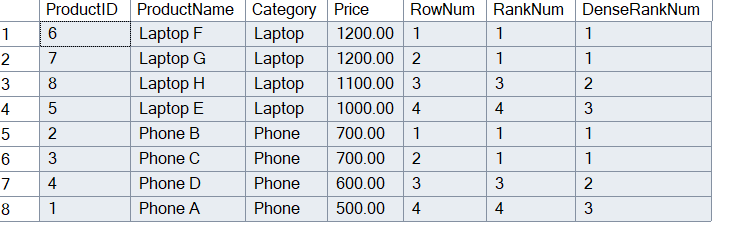
Price,

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

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

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

FROM Products;



**USING ROW\_NUMBER():**WITH RankedProducts AS (

SELECT \*,

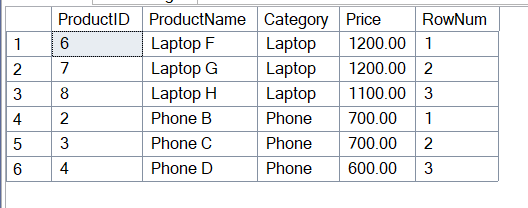
ROW\_NUMBER() OVER(PARTITION BY Category ORDER BY Price DESC) AS RowNum

FROM Products

)

SELECT \* FROM RankedProducts

WHERE RowNum<=3;



**USING RANK():**

WITH RankedProducts AS(

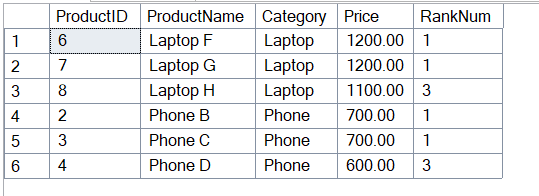
SELECT \*,

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

FROM Products)

SELECT \* FROM RankedProducts

WHERE RankNum<=3;



**USING DENSE\_RANK():**

WITH RankedProducts AS(

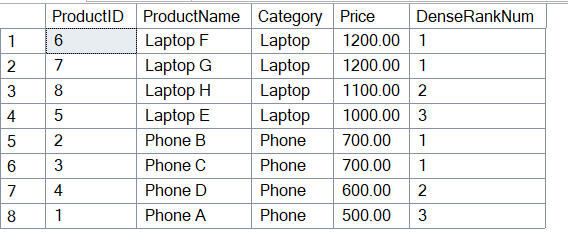
SELECT \*,

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

FROM Products)

SELECT \* FROM RankedProducts

WHERE DenseRankNum<=3;



**4. SQL Exercise:Stored Procedure**

***Exercise 1: Create a Stored Procedure:***

CREATE TABLE Departments(

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100));

CREATE TABLE Employees(

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE);

INSERT INTO Departments(DepartmentID, DepartmentName) VALUES

(1,'HR'),

(2,'Finance'),

(3,'IT'),

(4,'Marketing');

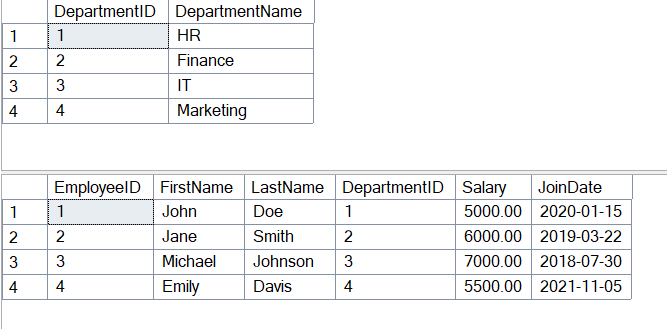
INSERT INTO Employees(EmployeeID,FirstName,LastName,DepartmentID,Salary,JoinDate) VALUES

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

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

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

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



CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

SELECT

E.EmployeeID,

E.FirstName,

E.LastName,

E.Salary,

E.JoinDate,

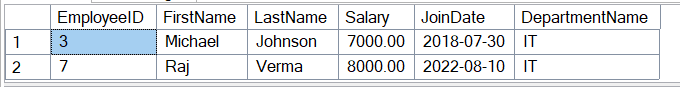
D.DepartmentName

FROM Employees E

INNER JOIN Departments D ON E.DepartmentID=D.DepartmentID

WHERE E.DepartmentID=@DepartmentID;

END;



CREATE PROCEDURE sp\_InsertEmployee

@EmployeeID INT,

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

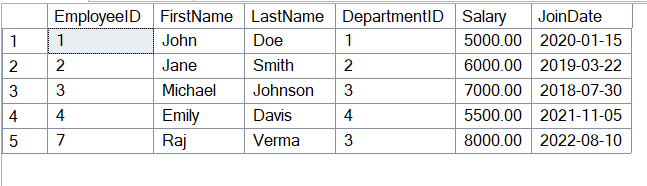
AS

BEGIN

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

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

END;



***Exercise 5: Return Data From Stored Procedure:***

CREATE PROCEDURE sp\_CountEmployeesInDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

EXEC sp\_CountEmployeesInDepartment @DepartmentID = 3;

