**Exercise 1: Ranking and Window Functions**

Goal: Use ROW\_NUMBER(), RANK(), DENSE\_RANK(), OVER(), and PARTITION BY

**Solution**

CREATE TABLE products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

category VARCHAR(50),

price DECIMAL(10, 2)

);

INSERT INTO products (product\_id, product\_name, category, price) VALUES

(1, 'Laptop A', 'Electronics', 1000.00),

(2, 'Laptop B', 'Electronics', 950.00),

(3, 'Laptop C', 'Electronics', 1000.00),

(4, 'Phone A', 'Electronics', 500.00),

(5, 'T-shirt A', 'Clothing', 30.00),

(6, 'T-shirt B', 'Clothing', 25.00),

(7, 'T-shirt C', 'Clothing', 30.00),

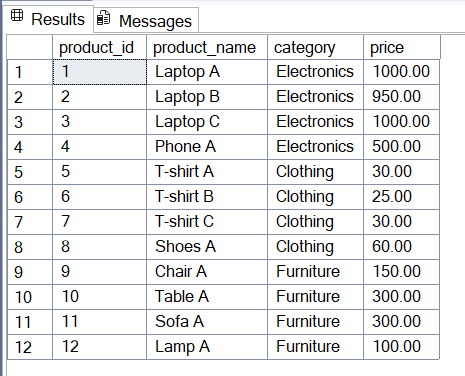
(8, 'Shoes A', 'Clothing', 60.00),

(9, 'Chair A', 'Furniture', 150.00),

(10, 'Table A', 'Furniture', 300.00),

(11, 'Sofa A', 'Furniture', 300.00),

(12, 'Lamp A', 'Furniture', 100.00);



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

Steps:

1. Use ROW\_NUMBER() to assign a unique rank within each category.

SELECT \*

FROM (

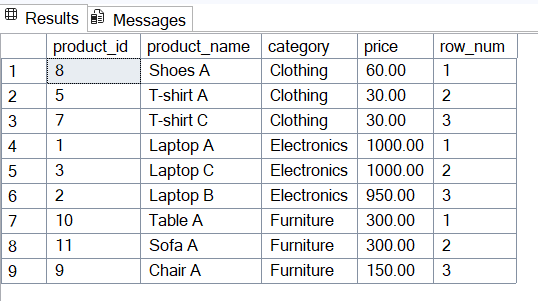
SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY category ORDER BY price DESC) AS row\_num

FROM products

) AS ranked

WHERE row\_num <= 3;



2.Use RANK() and DENSE\_RANK() to compare how ties are handled

SELECT \*

FROM (

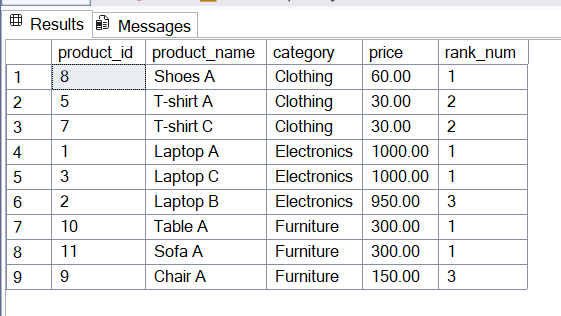
SELECT \*,

RANK() OVER (PARTITION BY category ORDER BY price DESC) AS rank\_num

FROM products

) AS ranked

WHERE rank\_num <= 3;



3.

SELECT \*

FROM (

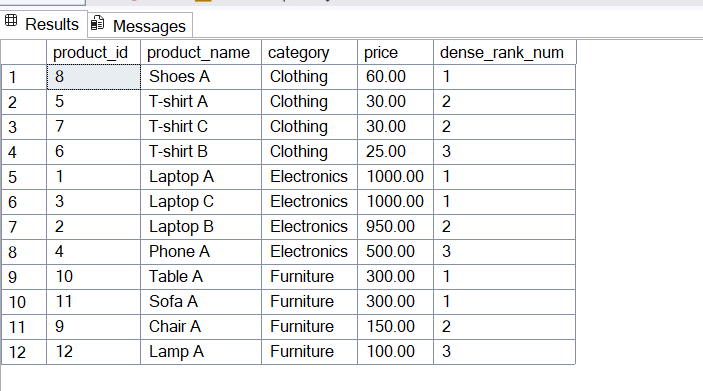
SELECT \*,

DENSE\_RANK() OVER (PARTITION BY category ORDER BY price DESC) AS dense\_rank\_num

FROM products

) AS ranked

WHERE dense\_rank\_num <= 3;



Stored Procedure

**Exercise 1: Create a Stored Procedure**

Goal: Create a stored procedure to retrieve employee details by department.

Steps:

1. Define the stored procedure with a parameter for DepartmentID.

2. Write the SQL query to select employee details based on the DepartmentID.

3. Create a stored procedure named `sp\_InsertEmployee` with the following code

**Solution**

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1), -- Auto-increment ID

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

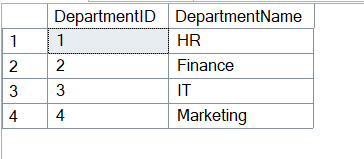
INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');



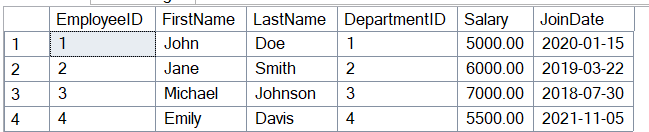
INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate) VALUES

('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

e.EmployeeID,

e.FirstName,

e.LastName,

d.DepartmentName,

e.Salary,

e.JoinDate

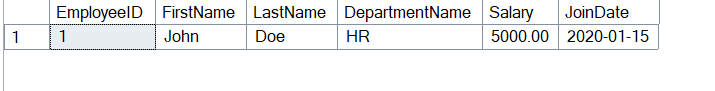
FROM Employees e

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

WHERE e.DepartmentID = @DeptID;

END;

EXEC sp\_GetEmployeesByDepartment @DeptID = 1;



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 = 'Robert',

@LastName = 'King',

@DepartmentID = 2,

@Salary = 6200.00,

@JoinDate = '2022-06-15';

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

Goal: Create a stored procedure that returns the total number of employees in a

department.

Steps:

1. Define the stored procedure with a parameter for DepartmentID.

2. Write the SQL query to count the number of employees in the specified department.

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

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SELECT

COUNT(\*) AS EmployeeCount

FROM

Employees

WHERE

DepartmentID = @DepartmentID;

END;

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 3;

