**Exercise 1: Ranking and Window Functions**

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

Scenario:

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

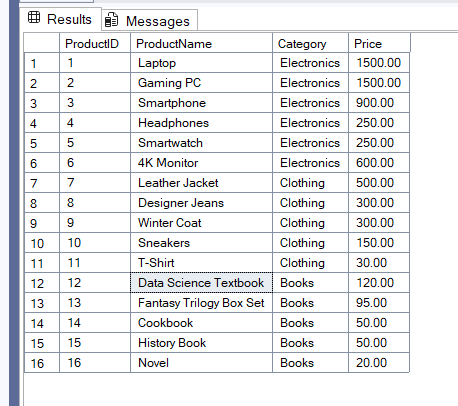
Steps:

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

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

3. Use PARTITION BY Category and ORDER BY Price DESC.

DATA:



1) a)Find the top 3 most expensive products in each category using ROW\_NUMBER()

WITH RankedProducts AS (

SELECT

ProductName,

Category,

Price,

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

FROM

dbo.Products

)

SELECT

Category,

ProductName,

Price,

RowNum

FROM

RankedProducts

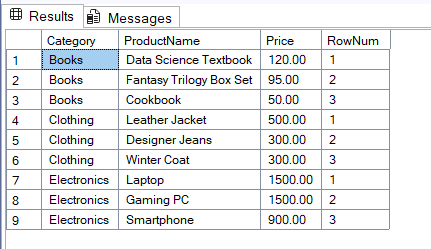
WHERE

RowNum <= 3

ORDER BY

Category,

RowNum;



1)b)Find the top 3 most expensive products in each category using RANK()

WITH RankedProducts AS (

SELECT

ProductName,

Category,

Price,

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

FROM

dbo.Products

)

SELECT

Category,

ProductName,

Price,

Rnk

FROM

RankedProducts

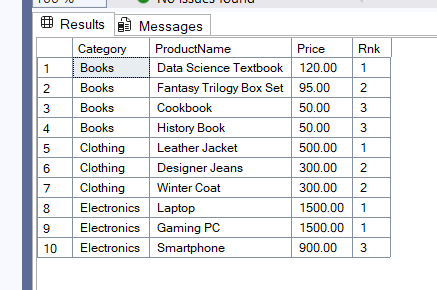
WHERE

Rnk <= 3

ORDER BY

Category,

Rnk;



1)c) Find the top 3 most expensive products in each category using DENSE\_RANK()

WITH RankedProducts AS (

SELECT

ProductName,

Category,

Price,

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

FROM

dbo.Products

)

SELECT

Category,

ProductName,

Price,

DenseRnk

FROM

RankedProducts

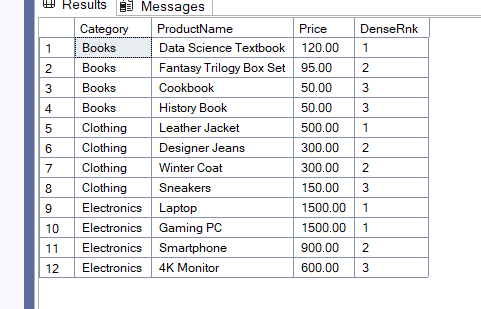
WHERE

DenseRnk <= 3

ORDER BY

Category,

DenseRnk;



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

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

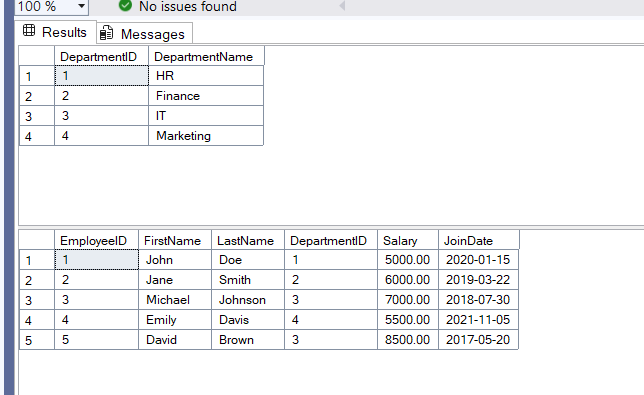
Steps:

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:

DATA



IF OBJECT\_ID('sp\_InsertEmployee', 'P') IS NOT NULL

DROP PROCEDURE sp\_InsertEmployee;

GO

CREATE PROCEDURE sp\_InsertEmployee

@EmployeeID INT,

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

AS

BEGIN

SET NOCOUNT ON;

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

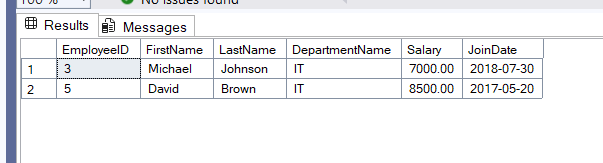
VALUES (@EmployeeID, @FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);

PRINT 'New employee inserted successfully.';

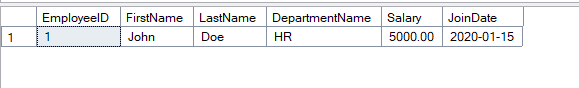
END;

GO

Test Case 1: Get all employees from the IT Department (ID = 3)



Test Case 2: Get all employees from the HR Department (ID = 1)



EXEC sp\_GetEmployeesByDepartment @DepartmentID = 4;

EXEC sp\_InsertEmployee

@EmployeeID = 6,

@FirstName = 'Sarah',

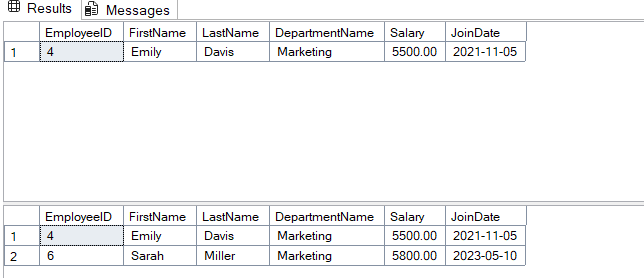
@LastName = 'Miller',

@DepartmentID = 4,

@Salary = 5800.00,

@JoinDate = '2023-05-10';

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 4;



3)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.

-- Step 0: Create and populate the database objects (if not already done)

-- Drop tables if they exist to ensure a clean start

IF OBJECT\_ID('dbo.Employees', 'U') IS NOT NULL

DROP TABLE dbo.Employees;

GO

IF OBJECT\_ID('dbo.Departments', 'U') IS NOT NULL

DROP TABLE dbo.Departments;

GO

-- Create the tables

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

GO

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

);

GO

-- Insert sample data

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

GO

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

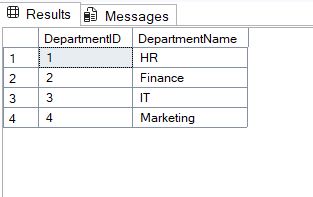
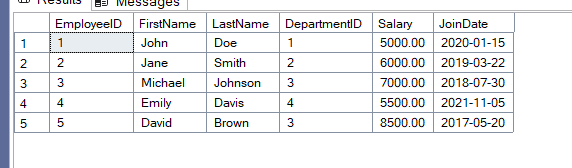
(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(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'),

(5, 'David', 'Brown', 3, 8500.00, '2017-05-20');



-- Drop the procedure if it already exists to avoid errors on re-run

IF OBJECT\_ID('sp\_GetEmployeeCountByDepartment', 'P') IS NOT NULL

DROP PROCEDURE sp\_GetEmployeeCountByDepartment;

GO

-- Create the stored procedure

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

-- 1. Define the input parameter for DepartmentID

@DepartmentID INT

AS

BEGIN

SET NOCOUNT ON;

-- 2. Write the SQL query to count employees and return it as a result set

SELECT

COUNT(EmployeeID) AS TotalEmployees

FROM

Employees

WHERE

DepartmentID = @DepartmentID;

END;

GO

-- Get the number of employees in the IT department (ID = 3)

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 3;

-- Get the number of employees in the HR department (ID = 1)

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 1;

