**SuperSet ID- 6362191  
SQL EXERCISE1: ADVANCED CONCEPTS**

Exercise 1: Ranking and Window Functions

Code:

-- Create Customers table

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name NVARCHAR(100),

Region NVARCHAR(50)

);

-- Create Products table

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName NVARCHAR(100),

Category NVARCHAR(50),

Price DECIMAL(10, 2)

);

-- Create Orders table

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Create OrderDetails table

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Insert into Customers

INSERT INTO Customers VALUES

(1, 'Alice', 'North'),

(2, 'Bob', 'South'),

(3, 'Charlie', 'East'),

(4, 'Diana', 'West');

-- Insert into Products

INSERT INTO Products VALUES

(101, 'Laptop', 'Electronics', 1500),

(102, 'Phone', 'Electronics', 800),

(103, 'TV', 'Electronics', 1200),

(201, 'Desk', 'Furniture', 300),

(202, 'Chair', 'Furniture', 150);

-- Insert into Orders

INSERT INTO Orders VALUES

(1001, 1, '2025-01-05'),

(1002, 2, '2025-01-08'),

(1003, 1, '2025-01-10'),

(1004, 3, '2025-01-15');

-- Insert into OrderDetails

INSERT INTO OrderDetails VALUES

(1, 1001, 101, 1),

(2, 1001, 102, 2),

(3, 1002, 201, 1),

(4, 1003, 101, 1),

(5, 1004, 103, 1);

SELECT \* FROM Customers;

SELECT \* FROM Products;

SELECT \* FROM Orders;

SELECT \* FROM OrderDetails;

SELECT

Category,

ProductName,

Price,

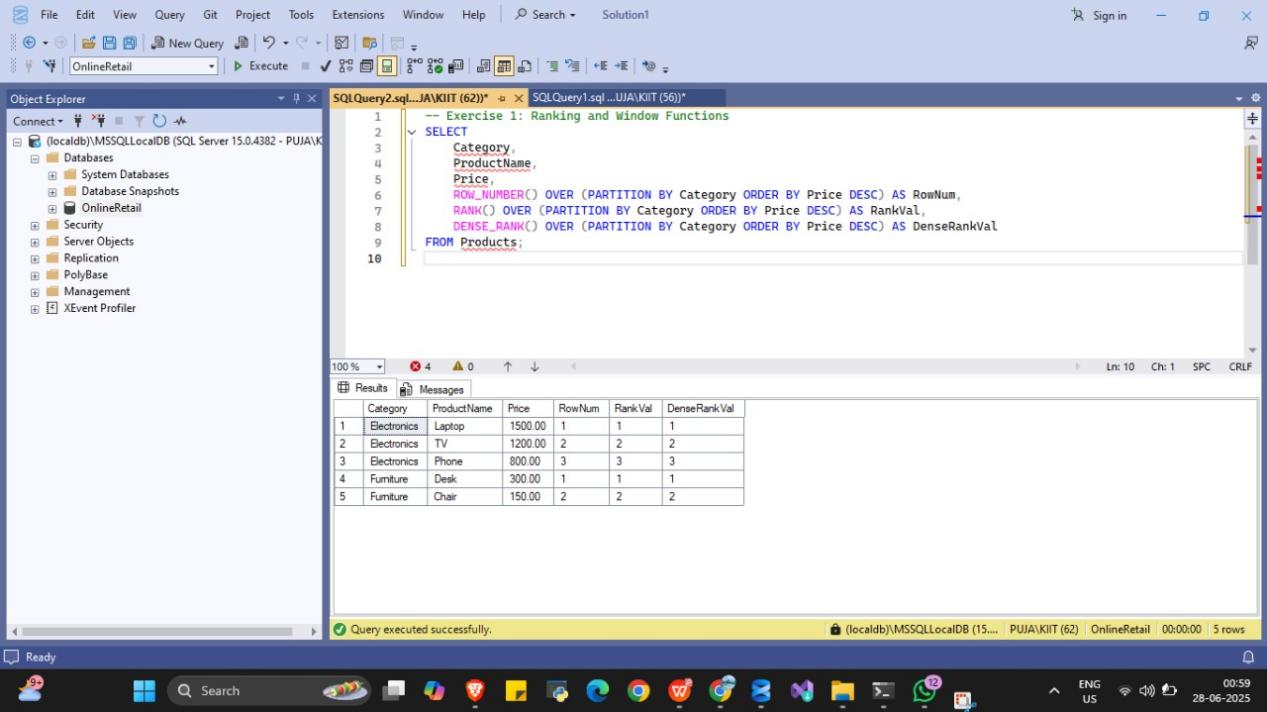
ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum,

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

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

FROM Products;

**Output:**



**SQL EXERCISE4: STORED PROCEDURE**

Exercise 1: Create a stored procedure

Code:

-- Step 1: Use the database (already created earlier)

USE EmployeeManagement;

GO

-- Step 2: Drop the stored procedure if it exists

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

DROP PROCEDURE sp\_InsertEmployee;

GO

-- Step 3: Drop the Employees table if it exists

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

DROP TABLE Employees;

GO

-- Step 4: Drop the Departments table if it exists

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

DROP TABLE Departments;

GO

-- Step 5: Recreate Departments table

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

GO

-- Step 6: Recreate Employees table

CREATE TABLE Employees (

EmployeeID INT IDENTITY(1,1) PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

GO

-- Step 7: Insert sample data into Departments

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

GO

-- Step 8: Insert sample data into Employees

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

-- Step 9: Create stored procedure

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

-- Step 10: Test the stored procedure

EXEC sp\_InsertEmployee

@FirstName = 'Alice',

@LastName = 'Walker',

@DepartmentID = 1,

@Salary = 5200.00,

@JoinDate = '2022-06-01';

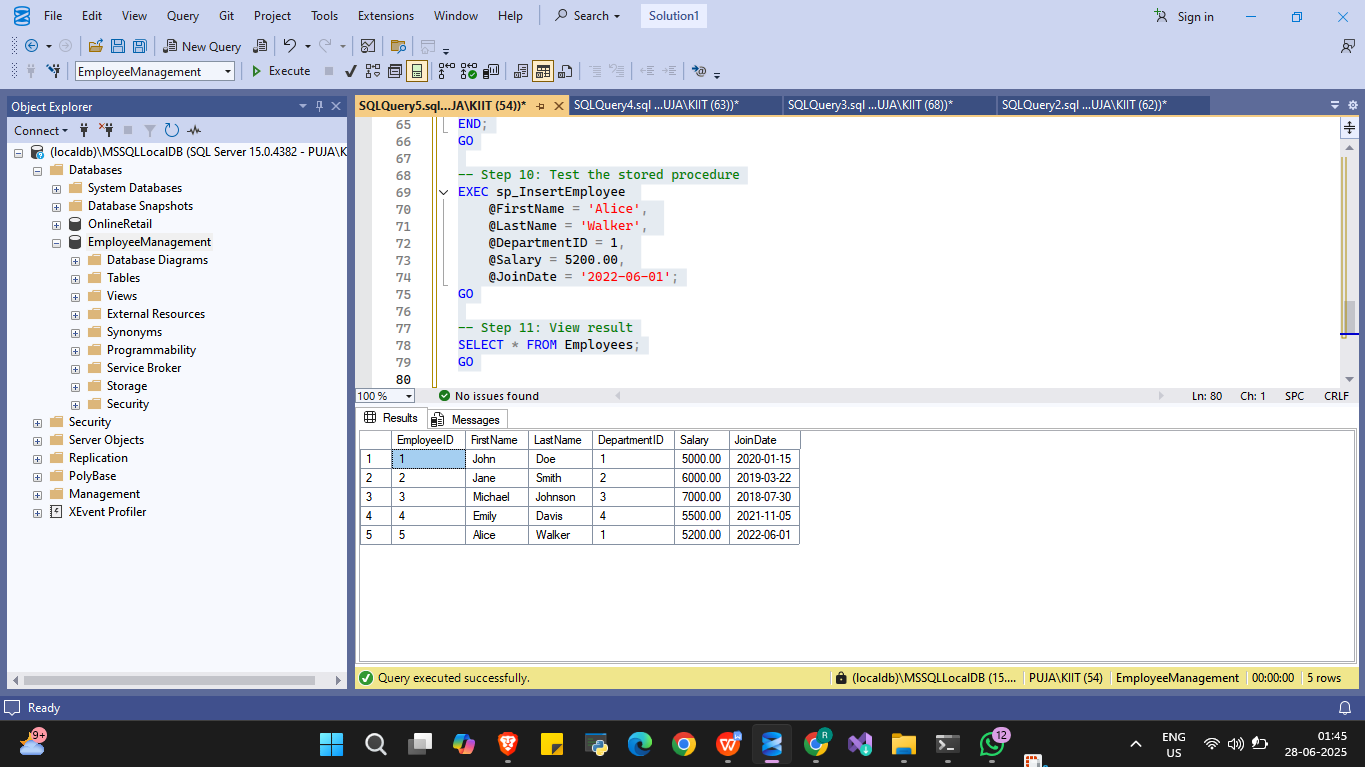
GO

-- Step 11: View result

SELECT \* FROM Employees;

GO

Output



**SQL EXERCISE4: STORED PROCEDURE**

Exercise 5: Return data from a stored procedure

Code:

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

DROP PROCEDURE sp\_GetEmployeeCountByDepartment;

GO

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

GO

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 1;

GO

Output:

