Advanced SQL Exercises for Online Retail Store

**Exercise 1: Ranking and Window Functions**

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

DROP TABLE Products;

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName NVARCHAR(100),

Category NVARCHAR(50),

Price DECIMAL(10, 2)

);

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

VALUES

(1, 'Laptop', 'Electronics', 1200.00),

(2, 'Smartphone', 'Electronics', 999.00),

(3, 'Tablet', 'Electronics', 850.00),

(4, 'Headphones', 'Electronics', 250.00),

(5, 'Shoes', 'Fashion', 200.00),

(6, 'Jacket', 'Fashion', 150.00),

(7, 'Watch', 'Fashion', 300.00),

(8, 'Bag', 'Fashion', 180.00),

(9, 'T-shirt', 'Fashion', 100.00);

WITH RankedProducts AS (

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

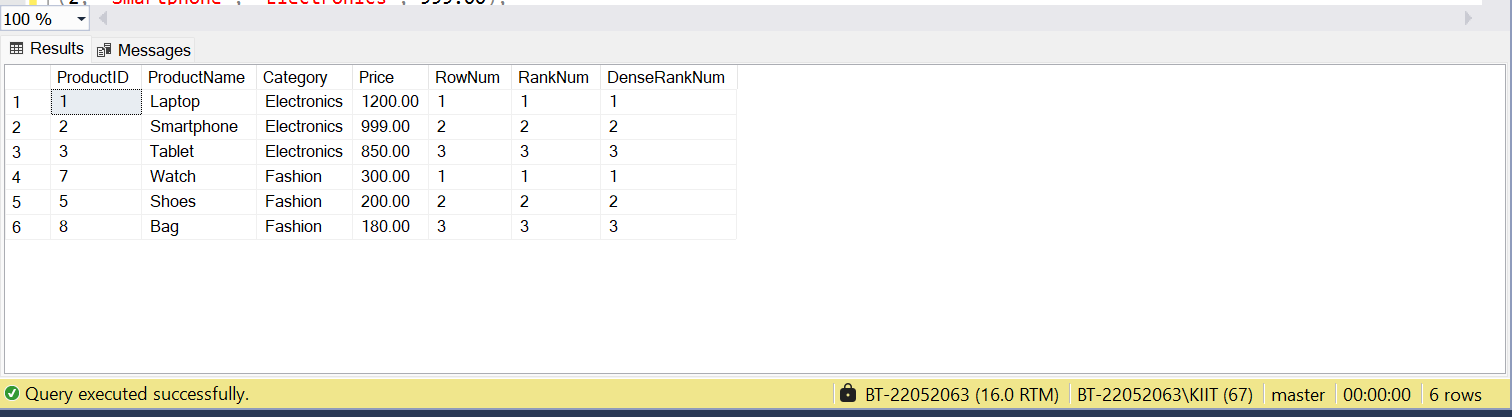
)

SELECT \*

FROM RankedProducts

WHERE RowNum <= 3

ORDER BY Category, Price DESC;

****

Employee Management System SQL Exercises

## Exercise 1: Create a Stored Procedure

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

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

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

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');

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

DROP PROCEDURE sp\_GetEmployeesByDepartment;

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID 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 = @DepartmentID;

END;

GO

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

DROP PROCEDURE sp\_InsertEmployee;

GO

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 = 'Prajwal',

@LastName = 'Kumar',

@DepartmentID = 2,

@Salary = 8000.00,

@JoinDate = '2023-06-01';

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 2;

SELECT

E.EmployeeID,

E.FirstName,

E.LastName,

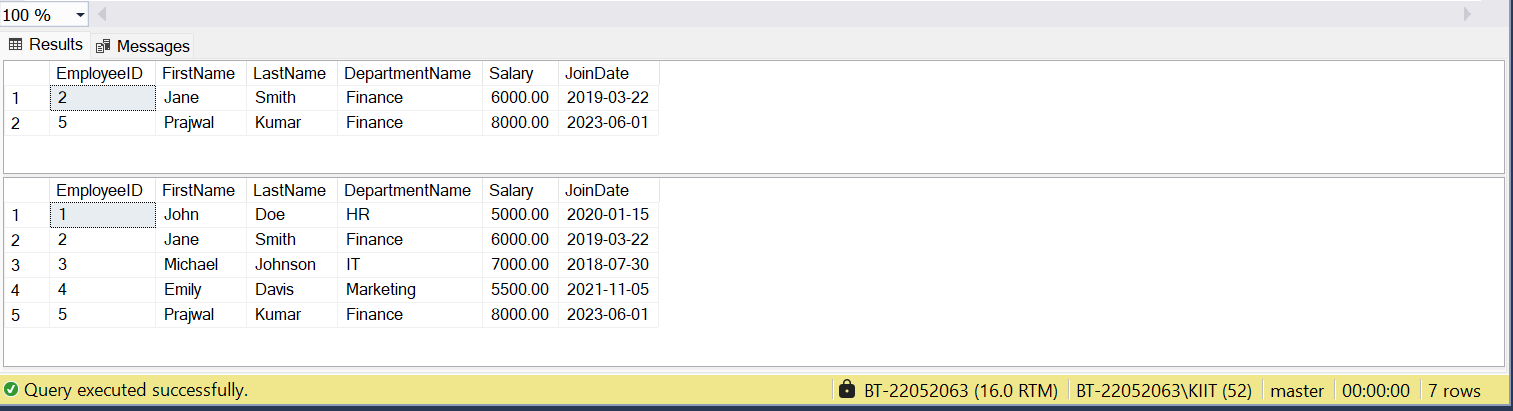
D.DepartmentName,

E.Salary,

E.JoinDate

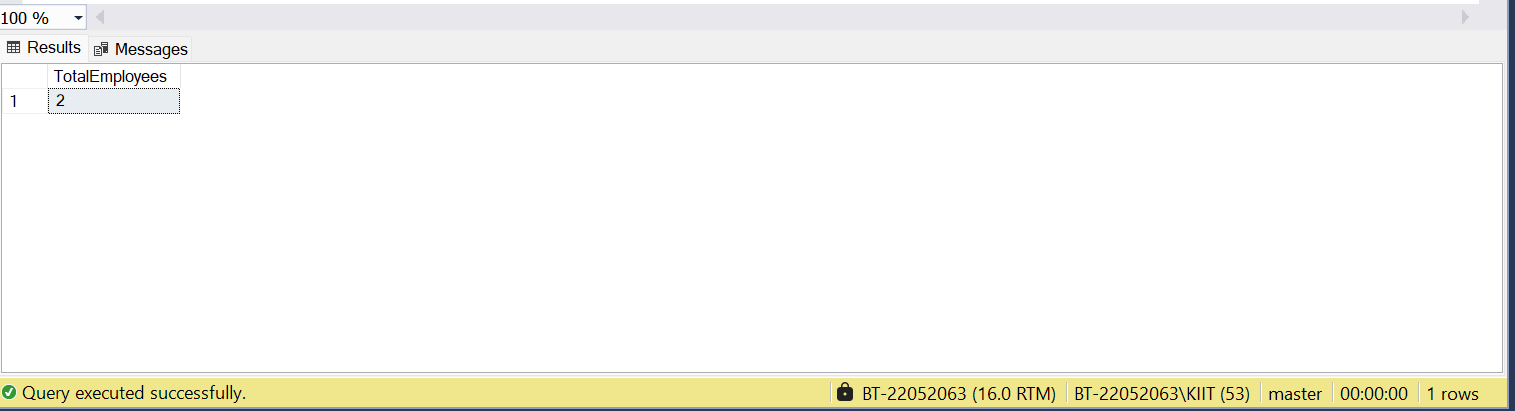
FROM Employees E

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

****

## Exercise 5: Return Data from a Stored Procedure

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 2;

****