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

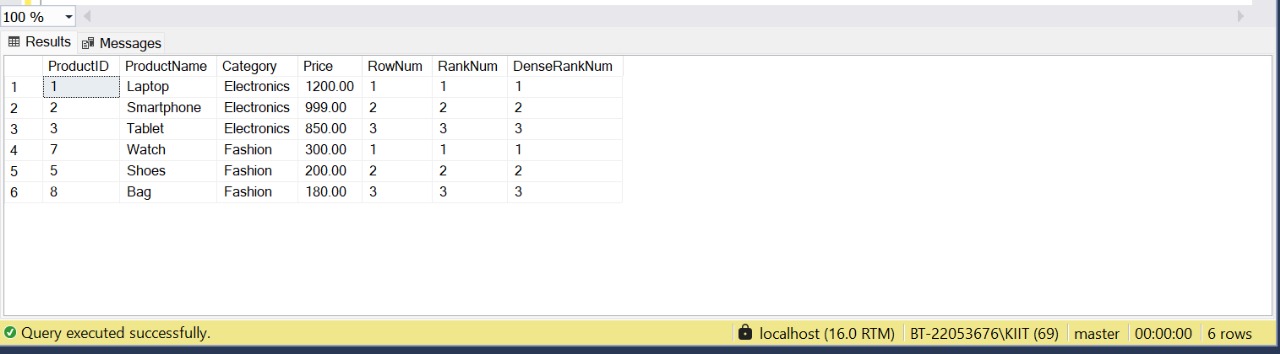
)

SELECT \*

FROM RankedProducts

WHERE RowNum <= 3

ORDER BY Category, Price DESC;



Employee Management System SQL Exercises

## Exercise 1: Create a Stored Procedure

-- 1. Drop tables if they already exist (in correct dependency order)

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

DROP TABLE dbo.Employees;

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

DROP TABLE dbo.Departments;

-- 2. Create Departments table

CREATE TABLE dbo.Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100) NOT NULL

);

-- 3. Create Employees table

CREATE TABLE dbo.Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT NOT NULL

FOREIGN KEY REFERENCES dbo.Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

-- 4. Insert departments data

INSERT INTO dbo.Departments (DepartmentID, DepartmentName)

VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

-- 5. Insert multiple employees in one go

INSERT INTO dbo.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');

-- 6. Verify the inserted data

SELECT

e.EmployeeID,

e.FirstName,

e.LastName,

e.DepartmentID,

d.DepartmentName,

e.Salary,

e.JoinDate

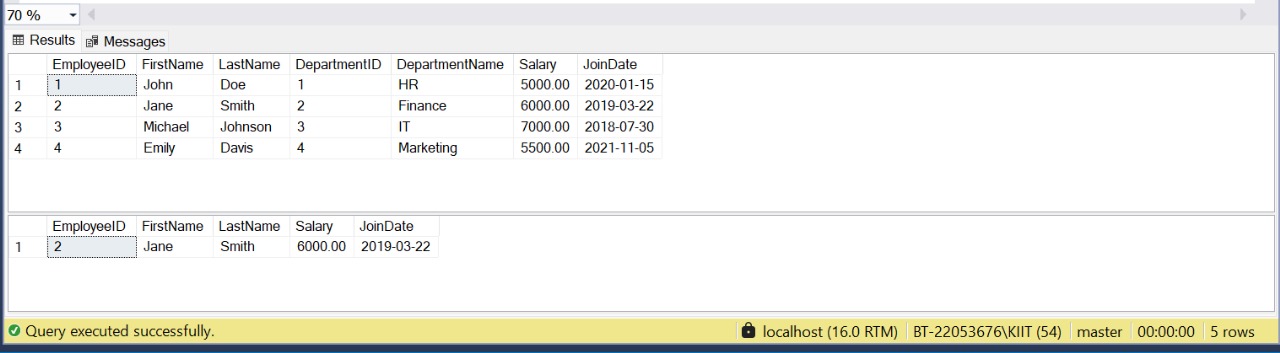
FROM dbo.Employees e

JOIN dbo.Departments d

ON e.DepartmentID = d.DepartmentID

ORDER BY e.EmployeeID;

EXEC dbo.sp\_GetEmployeesByDepartment @DepartmentID = 2;



## Exercise 5: Return Data from a Stored Procedure

-- 🧹 Step 1: Drop existing tables and procedure (if they exist)

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

IF OBJECT\_ID('dbo.sp\_ShowEmployeesAndCount','P') IS NOT NULL DROP PROCEDURE dbo.sp\_ShowEmployeesAndCount;

GO

-- 🏗️ Step 2: Create Departments table

CREATE TABLE dbo.Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100) NOT NULL

);

GO

-- 🏗️ Step 3: Create Employees table

CREATE TABLE dbo.Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT NOT NULL FOREIGN KEY REFERENCES dbo.Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

GO

-- 📝 Step 4: Insert sample data into Departments

INSERT INTO dbo.Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

GO

-- 👥 Step 5: Insert sample data into Employees

INSERT INTO dbo.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');

GO

-- 👨‍💼 Step 6: Create the stored procedure to display employees and their count for a department

CREATE PROCEDURE dbo.sp\_ShowEmployeesAndCount

@DepartmentID INT = NULL

AS

BEGIN

SET NOCOUNT ON;

-- Display all employees (optionally filtered by department)

SELECT

e.EmployeeID,

e.FirstName,

e.LastName,

e.DepartmentID,

d.DepartmentName,

e.Salary,

e.JoinDate

FROM dbo.Employees e

JOIN dbo.Departments d

ON e.DepartmentID = d.DepartmentID

WHERE (@DepartmentID IS NULL OR e.DepartmentID = @DepartmentID)

ORDER BY e.EmployeeID;

-- Display count of those employees

SELECT COUNT(\*) AS TotalEmployees

FROM dbo.Employees

WHERE (@DepartmentID IS NULL OR DepartmentID = @DepartmentID);

END;

GO

-- ▶️ Step 7: Execute the stored procedure

-- To show all employees

EXEC dbo.sp\_ShowEmployeesAndCount;

-- To show only Finance employees and count

EXEC dbo.sp\_ShowEmployeesAndCount @DepartmentID = 2;

