Name:-Ritesh Kumar Sahoo

GitHub Link:-[https://github.com/Riteshksahoo/Cognizant\_6364581\_Ritesh](#_top)CognizantDotNet

**-----------------------------------------------------------------------------------------**

**SQL Exercise - Advanced concepts  
  
Advanced SQL Exercises for Online Retail Store  
  
Exercise 1: Ranking and Window Functions  
  
INPUT:-**DROP TABLE IF EXISTS Products;

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

INSERT INTO Products VALUES

(1, 'Laptop Lenovo', 'Electronics', 950.00),

(2, 'Laptop MacBook', 'Electronics', 1250.00),

(3, 'Phone IPhone', 'Electronics', 1250.00),

(4, 'Tablet Samsung', 'Electronics', 650.00),

(5, 'Sofa Set', 'Furniture', 750.00),

(6, 'Dining Table', 'Furniture', 950.00),

(7, 'Chair', 'Furniture', 950.00),

(8, 'Bookshelf', 'Furniture', 550.00),

(9, 'T-Shirt', 'Clothing', 25.00),

(10, 'Jacket', 'Clothing', 95.00),

(11, 'Jeans', 'Clothing', 95.00),

(12, 'Shoes', 'Clothing', 60.00);

SELECT \*

FROM (

SELECT \*,

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

FROM Products

) AS Ranked

WHERE RowNum <= 3

ORDER BY Category, RowNum;

SELECT \*

FROM (

SELECT \*,

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

FROM Products

) AS Ranked

WHERE RankVal <= 3

ORDER BY Category, RankVal;

SELECT \*

FROM (

SELECT \*,

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

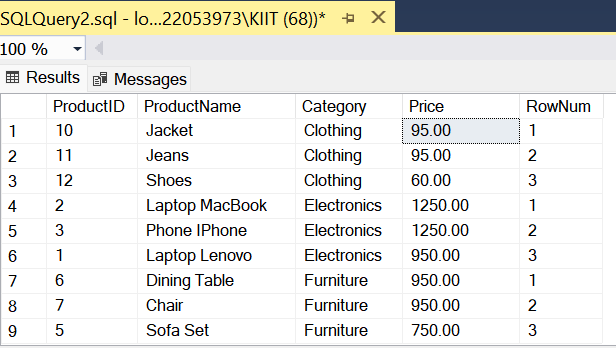
FROM Products

) AS Ranked

WHERE DenseRankVal <= 3

ORDER BY Category, DenseRankVal;

*Output in page no. 2*

**OUTPUT:-**   
 **-----------------------------------------------------------------------------------------**  
  
  
  
  
  
  
  
 **SQL Exercise - Advanced concepts  
  
SQL Exercise - Stored procedure**

**Exercise 1: Create a Stored Procedure  
  
INPUT:-**  
  
DROP TABLE IF EXISTS Employees;

GO

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(100),

Department VARCHAR(50),

Salary DECIMAL(10,2)

);

GO

INSERT INTO Employees (FullName, Department, Salary) VALUES

('Alice Johnson', 'IT', 75000.00),

('Bob Singh', 'HR', 55000.00),

('Charlie Verma', 'IT', 72000.00),

('Diana Mehta', 'Marketing', 60000.00);

GO

DROP PROCEDURE IF EXISTS sp\_GetEmployeesByDepartment;

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentName VARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

SELECT EmployeeID, FullName, Department, Salary

FROM Employees

WHERE Department = @DepartmentName;

END;

GO

DROP PROCEDURE IF EXISTS sp\_InsertEmployee;

GO

CREATE PROCEDURE sp\_InsertEmployee

@FullName VARCHAR(100),

@Department VARCHAR(50),

@Salary DECIMAL(10,2)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO Employees (FullName, Department, Salary)

VALUES (@FullName, @Department, @Salary);

END;

GO

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

DROP TABLE IF EXISTS Employees;

GO

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(100),

Department VARCHAR(50),

Salary DECIMAL(10,2)

);

GO

INSERT INTO Employees (FullName, Department, Salary) VALUES

('Alice Johnson', 'IT', 75000.00),

('Bob Singh', 'HR', 55000.00),

('Charlie Verma', 'IT', 72000.00),

('Diana Mehta', 'Marketing', 60000.00);

GO

DROP PROCEDURE IF EXISTS sp\_GetEmployeeCountByDepartment;

GO

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@Department VARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

SELECT COUNT(\*) AS EmployeeCount

FROM Employees

WHERE Department = @Department;

END;

GO

**-----------------------------------------------------------------------------------------  
  
ADDITIONAL IMPORTANT HANDS ON**