**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.   
  
  
SELECT \*

FROM (SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products

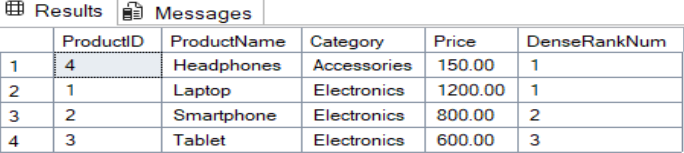
) AS Ranked

WHERE RowNum <= 3;  
  
**OUTPUT:**A table with text on it

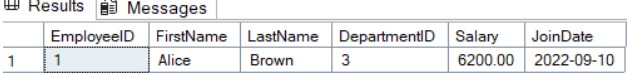
AI-generated content may be incorrect.

2. Use RANK() and DENSE\_RANK() to compare how ties are handled.   
  
SELECT \*FROM ( SELECT ProductID, ProductName, Category, Price, RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum FROM Products) AS RankedWHERE RankNum <= 3;  
  
**OUTPUT:**  
  
A screenshot of a computer

AI-generated content may be incorrect.

3. Use PARTITION BY Category and ORDER BY Price DESC.  
  
SELECT \*FROM ( SELECT ProductID, ProductName, Category, Price, DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankNum FROM Products) AS RankedWHERE DenseRankNum <= 3;  
  
**OUTPUT:**

**Exercise -2 Create a Stored Procedure**

DROP PROCEDURE IF EXISTS sp\_GetEmployeesByDepartment;GOCREATE PROCEDURE sp\_GetEmployeesByDepartment @DepartmentID INTASBEGIN SELECT E.EmployeeID, E.FirstName, E.LastName, E.Salary, E.JoinDate, D.DepartmentName FROM Employees E INNER JOIN Departments D ON E.DepartmentID = D.DepartmentID WHERE E.DepartmentID = @DepartmentID;END;GOEXEC sp\_GetEmployeesByDepartment @DepartmentID = 2;-- Drop if it already existsDROP PROCEDURE IF EXISTS sp\_InsertEmployee;GO-- Create the procedureCREATE PROCEDURE sp\_InsertEmployee @FirstName VARCHAR(50), @LastName VARCHAR(50), @DepartmentID INT, @Salary DECIMAL(10,2), @JoinDate DATEASBEGIN INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate) VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);END;GOEXEC sp\_InsertEmployee @FirstName = 'Alice', @LastName = 'Brown', @DepartmentID = 3, @Salary = 6200.00, @JoinDate = '2022-09-10';SELECT \* FROM Employees;  
  
**output :**  
  
  
  
  
**Exercise - 3 Return Data from a Stored Procedure**-- Drop if it already existsDROP PROCEDURE IF EXISTS sp\_CountEmployeesByDepartment;GO-- Create the procedureCREATE PROCEDURE sp\_CountEmployeesByDepartment @DepartmentID INTASBEGIN SELECT COUNT(\*) AS TotalEmployees FROM Employees WHERE DepartmentID = @DepartmentID;END;GOEXEC sp\_CountEmployeesByDepartment @DepartmentID = 2;  
  
**OUTPUT:**  
  
