Ex-1 Ranking and Window Functions

Step 1: Use ROW_NUMBER() to assign a unique rank

```
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;
```

Results | Messages

	ProductID	ProductName	Category	Price	RowNum
1	4	Headphones	Accessories	150.00	1
2	1	Laptop	Electronics	1200.00	1
3	2	Smartphone	Electronics	800.00	2
4	3	Tablet	Electronics	600.00	3

Step 2: Use RANK() to handle ties

```
SELECT *
FROM (
   SELECT
      ProductID,
      ProductName,
      Category,
      Price,
      RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum
   FROM Products
) AS Ranked
WHERE RankNum <= 3;
```

Results Messages

	ProductID	ProductName	Category	Price	RankNum
1	4	Headphones	Accessories	150.00	1
2	1	Laptop	Electronics	1200.00	1
3	2	Smartphone	Electronics	800.00	2
4	3	Tablet	Electronics	600.00	3

Step 3: Use DENSE_RANK() to handle ties

```
SELECT *
FROM (
   SELECT
      ProductID,
      ProductName,
      Category,
      Price,
      DENSE_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankNum
   FROM Products
) AS Ranked
WHERE DenseRankNum <= 3;
```

Results Messages

	ProductID	ProductName	Category	Price	DenseRankNum
1	4	Headphones	Accessories	150.00	1
2	1	Laptop	Electronics	1200.00	1
3	2	Smartphone	Electronics	800.00	2
4	3	Tablet	Electronics	600.00	3

EX-2 Create a Stored Procedure

```
DROP PROCEDURE IF EXISTS sp_GetEmployeesByDepartment;
GO
CREATE PROCEDURE sp_GetEmployeesByDepartment
    @DepartmentID INT
AS
BEGIN
    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;
GO

EXEC sp_GetEmployeesByDepartment @DepartmentID = 2;
```

Results Messages

	EmployeeID	FirstName	LastName	Salary	JoinDate	DepartmentName
1	2	Jane	Smith	6000.00	2019-03-22	Finance

```
-- Drop if it already exists
DROP PROCEDURE IF EXISTS sp_InsertEmployee;
GO
```

```
-- Create the 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

EXEC sp_InsertEmployee
    @FirstName = 'Alice',
    @LastName = 'Brown',
    @DepartmentID = 3,
    @Salary = 6200.00,
    @JoinDate = '2022-09-10';
SELECT * FROM Employees;
```

Results | Messages

	EmployeeID	FirstName	LastName	DepartmentID	Salary	JoinDate
1	1	Alice	Brown	3	6200.00	2022-09-10

EX-3 Return Data from a Stored Procedure

```
-- Drop if it already exists
DROP PROCEDURE IF EXISTS sp_CountEmployeesByDepartment;
GO

-- Create the procedure
CREATE PROCEDURE sp_CountEmployeesByDepartment
    @DepartmentID INT
AS
BEGIN
    SELECT
        COUNT(*) AS TotalEmployees
    FROM Employees
    WHERE DepartmentID = @DepartmentID;
END;
GO

EXEC sp_CountEmployeesByDepartment @DepartmentID = 2;
```