Week-2 Hands-on-Exerise

Advance SQL Server

SQL EXERCISE - Advanced Concepts

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.

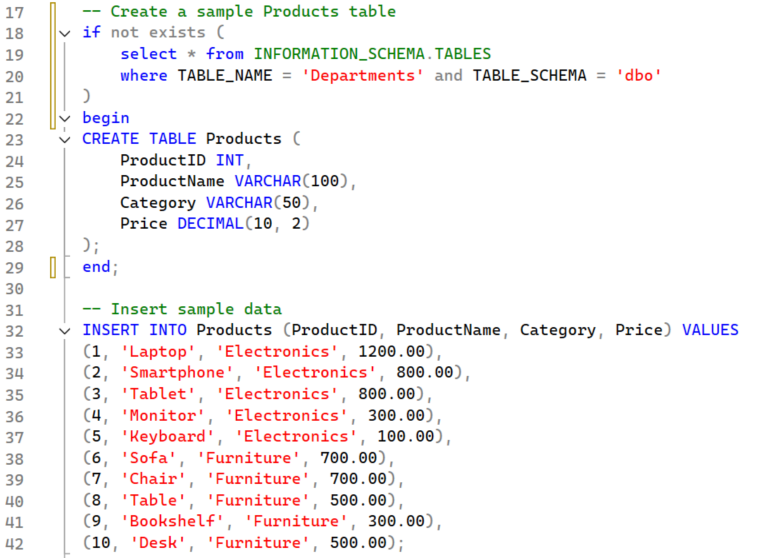
Steps:

1. Use ROW\_NUMBER() to assign a unique rank within each category.

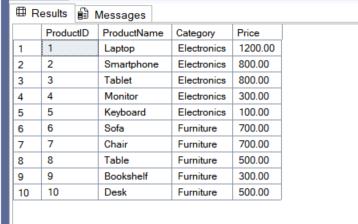
2. Use RANK() and DENSE\_RANK() to compare how ties are handled.

3. Use PARTITION BY Category and ORDER BY Price DESC.

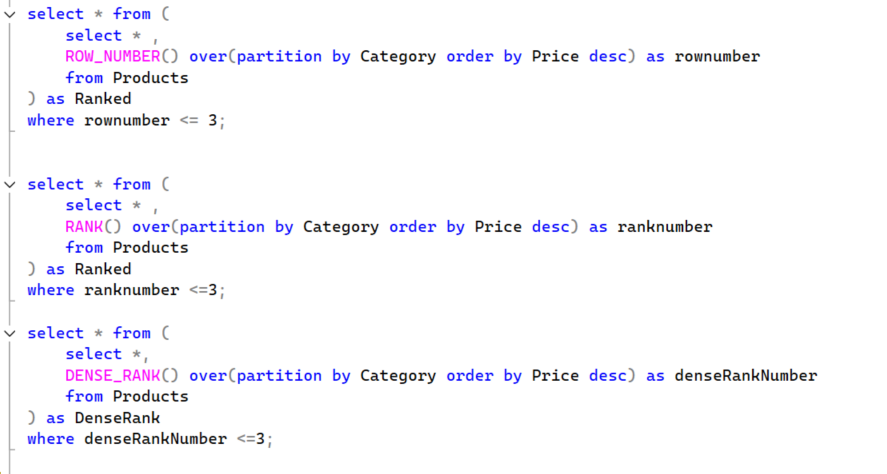
Sql commands for creating and inserting random data:-



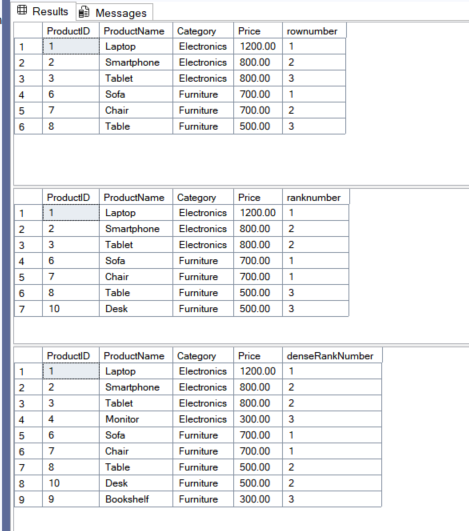
Following out show the database:-



Following code is for displaying ROW\_NUMBER() , RANK() and DENSE\_RANK()



The output of above code including ROW\_NUMBER() , RANK() and DENSE\_RANK():-



SQL EXERCISE - Stored Procedure

Exercise 1: Create a Stored Procedure

Goal: Create a stored procedure to retrieve employee details by department.

Steps:

1. Define the stored procedure with a parameter for DepartmentID.

2. Write the SQL query to select employee details based on the DepartmentID.

3. Create a stored procedure named `sp\_InsertEmployee` with the following code:

CREATE PROCEDURE sp\_InsertEmployee

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

AS

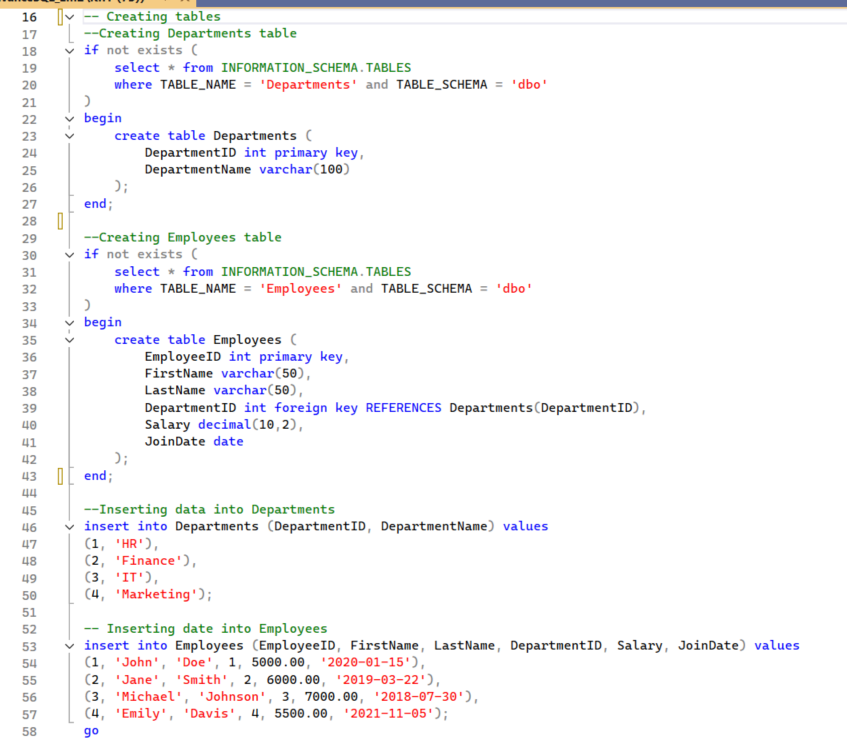
BEGIN

INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);

END;

Creating the database schema and inserting the data into it



1. Creating stored procedure ‘sp\_GetEmployeesByDepartment’ which takes DepartmentID and display the employee record match with the DepartmentID
2. Creating stored procedure named ‘sp\_insertEmployee’ with the following code:-

REATE PROCEDURE sp\_InsertEmployee

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

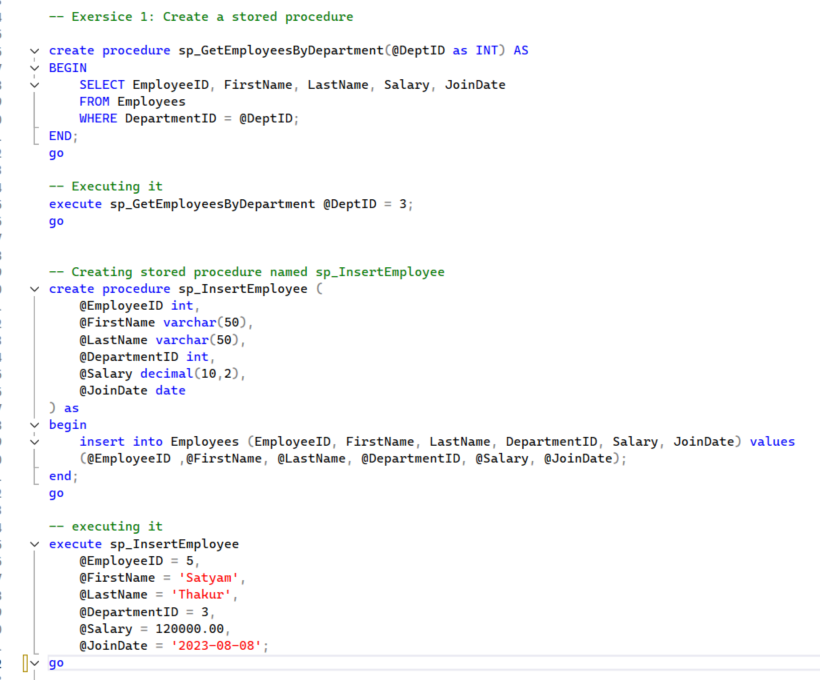
AS

BEGIN

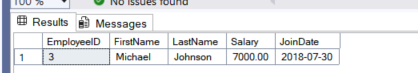
INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);

END;



Output of stored procedure ‘sp\_GetEmployeesByDepartment’ where the DepartmentId = 3 :-



Output of stored procedure ‘sp\_insertEmployee’

With execute sp\_InsertEmployee

@EmployeeID = 5,

@FirstName = 'Satyam',

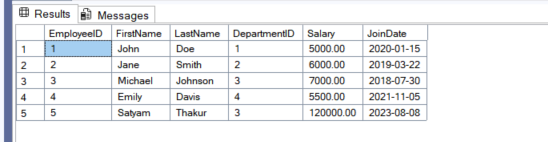
@LastName = 'Thakur',

@DepartmentID = 3,

@Salary = 120000.00,

@JoinDate = '2023-08-08'

:-



Exercise 5: Return Data from a Stored Procedure:-

Goal: Create a stored procedure that returns the total number of employees in a

department.

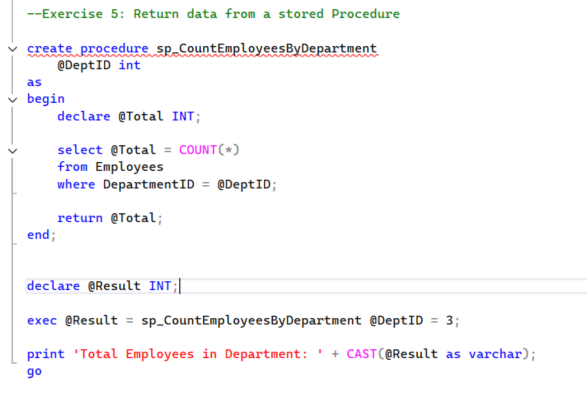
Steps:

1. Define the stored procedure with a parameter for DepartmentID.

2. Write the SQL query to count the number of employees in the specified department.

3. Save the stored procedure by executing the Stored procedure content

Screenshot of Creation of stored procedure:-



Output:-

