# Week – 2 Hands On Exercise( Advance Sql)

Exercise 1: Ranking and Window Functions

1. A ) Creating a table products

**CREATE TABLE products (**

**product\_id INT PRIMARY KEY,**

**product\_name VARCHAR(100),**

**category VARCHAR(50),**

**price DECIMAL(10,2)**

**);**

**b)** Inserting values

**INSERT INTO products VALUES**

**(1, 'Premium Laptop', 'Electronics', 1299.99),**

**(2, 'Wireless Headphones', 'Electronics', 199.99),**

**(3, 'Smartphone Pro', 'Electronics', 1099.99),**

**(4, 'Bluetooth Speaker', 'Electronics', 199.99),**

**(5, 'Organic Cotton T-Shirt', 'Clothing', 29.99),**

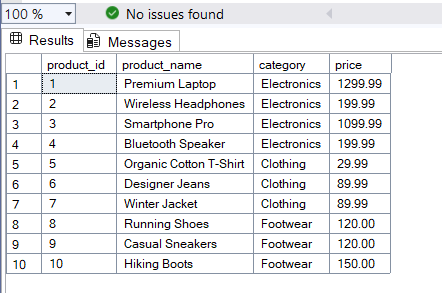
**(6, 'Designer Jeans', 'Clothing', 89.99),**

**(7, 'Winter Jacket', 'Clothing', 89.99),**

**(8, 'Running Shoes', 'Footwear', 120.00),**

**(9, 'Casual Sneakers', 'Footwear', 120.00),**

**(10, 'Hiking Boots', 'Footwear', 150.00);**



/\* using random\_number \*/

**SELECT**

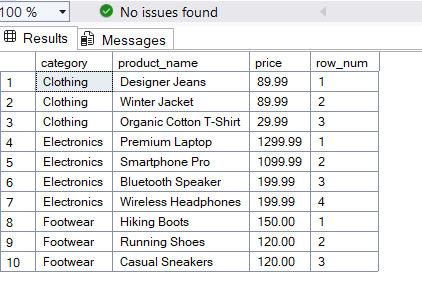
**category,**

**product\_name,**

**price,**

**ROW\_NUMBER() OVER (PARTITION BY category ORDER BY price DESC) AS row\_num**

**FROM products;**

****

/\* using rank and dense\_rank ,partition by with desc \*/

**SELECT**

**emp\_name,**

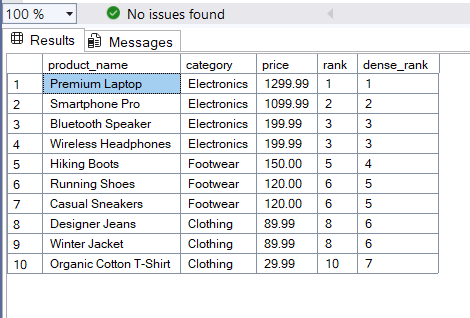
**emp\_work,**

**RANK() OVER (ORDER BY emp\_work DESC) AS rank\_a,**

**DENSE\_RANK() OVER (ORDER BY emp\_work DESC) AS dense\_ranke**

**FROM employee**

**ORDER BY emp\_work DESC;**

****

Stored Procedure

Exercise 1: Create a Stored Procedure

1. Creating procedure details by keeping dept\_id as parameter

**CREATE PROCEDURE details @DepartmentID INT**

**AS**

**BEGIN**

**SELECT**

**e.EmployeeID,**

**e.FirstName,**

**e.LastName,**

**d.DepartmentName,**

**e.Salary,**

**e.JoinDate,**

**DATEDIFF(YEAR, e.JoinDate, GETDATE()) AS YearsOfService**

**FROM**

**Employees e**

**INNER JOIN**

**Departments d ON e.DepartmentID = d.DepartmentID**

**WHERE**

**e.DepartmentID = @DepartmentID**

**ORDER BY**

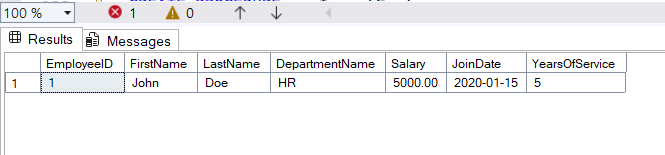
**e.LastName, e.FirstName;**

**END;**

**GO**

/\* executing the procedure \*/

**EXEC details @DepartmentID = 1;**

****

1. Creating a procedure sp\_InsertEmployee

/\* executing the created procedure \*/

**CREATE OR ALTER PROCEDURE sp\_InsertEmployee**

**@EmployeeID INT,**

**@FirstName VARCHAR(50),**

**@LastName VARCHAR(50),**

**@DepartmentID INT,**

**@Salary DECIMAL(10,2),**

**@JoinDate DATE**

**AS**

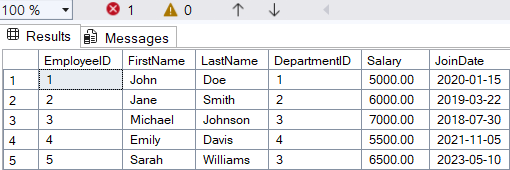
**BEGIN**

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

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

**END;**

**GO**

****

**/\* executing the procedure \*/**

**EXEC sp\_InsertEmployee**

**@EmployeeID = 6,**

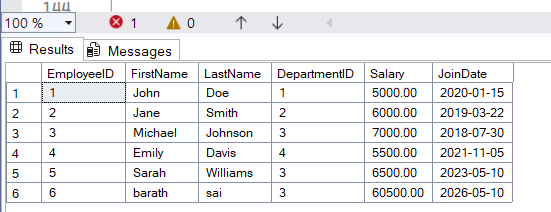
**@FirstName = 'barath',**

**@LastName = 'sai',**

**@DepartmentID = 3,**

**@Salary = 60500.00,**

**@JoinDate = '2026-05-10';**



Exercise 5: Return Data from a Stored Procedure

creating a stored procedure for giving count of each employee group by department

**CREATE OR ALTER PROCEDURE sp\_GetEmployeeCount**

**@DepartmentID INT**

**AS**

**BEGIN**

**DECLARE @EmployeeCount INT;**

**SELECT @EmployeeCount = COUNT(\*)**

**FROM Employees**

**WHERE DepartmentID = @DepartmentID;**

**SELECT**

**d.DepartmentName,**

**@EmployeeCount AS EmployeeCount**

**FROM**

**Departments d**

**WHERE**

**d.DepartmentID = @DepartmentID;**

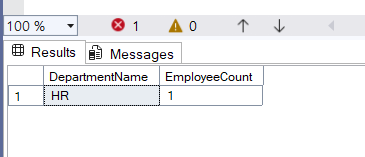
**END;**

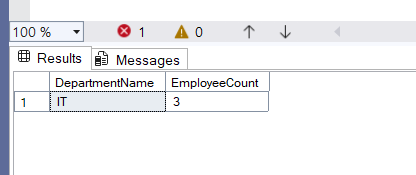
**GO**

Executing the stored procedure

**EXEC sp\_GetEmployeeCount @DepartmentID = 1;**

**ExEC sp\_GetEmployeeCount @DepartmentID=3;**

****

****