**Exercise 1: Ranking and Window Functions**

**SQL CODE :**

CREATE TABLE Employees

(

EmployeeID INT PRIMARY KEY,

Name VARCHAR(50),

Department VARCHAR(50),

Salary INT

);

INSERT INTO Employees (EmployeeID, Name, Department, Salary) VALUES

(101, 'Alice', 'IT', 60000),

(102, 'Bob', 'HR', 55000),

(103, 'Charlie', 'IT', 60000),

(104, 'David', 'HR', 50000),

(105, 'Eve', 'IT', 70000),

(106, 'Frank', 'Sales', 50000),

(107, 'Grace', 'Sales', 55000);

SELECT \* FROM Employees;

SELECT

EmployeeID,

Name,

Department,

Salary,

RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS SalaryRank

FROM Employees;

SELECT

EmployeeID,

Name,

Department,

Salary,

DENSE\_RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS DenseSalaryRank

FROM Employees;

SELECT

EmployeeID,

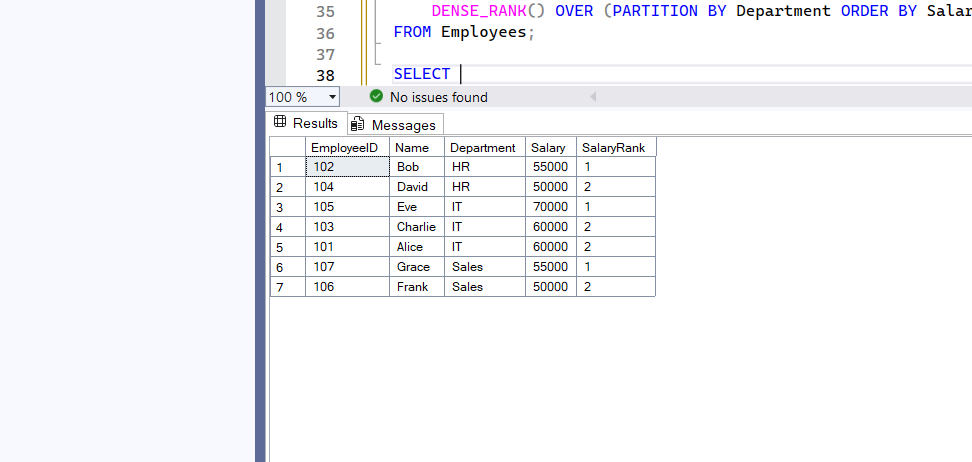
Name,

Department,

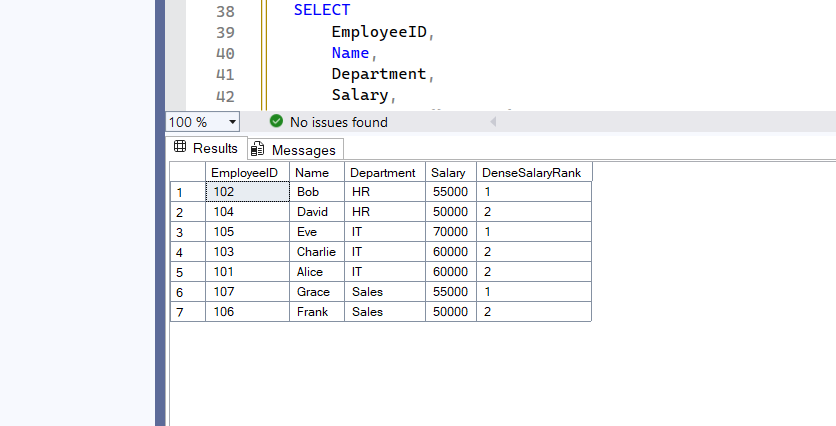
Salary,

ROW\_NUMBER() OVER (PARTITION BY Department ORDER BY Salary DESC) AS RowNum

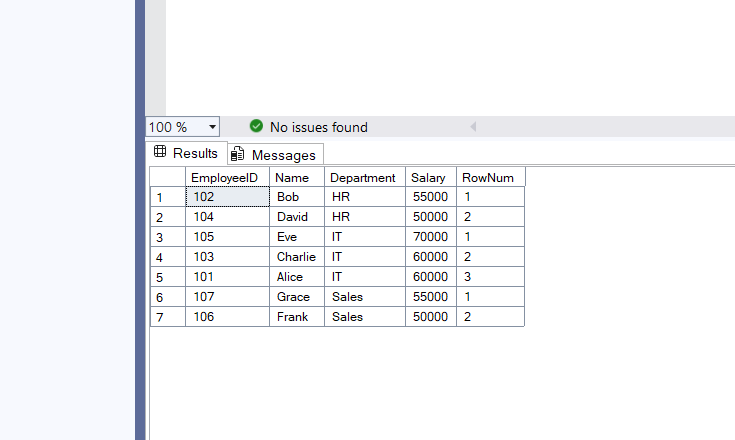
FROM Employees;

**OUTPUT FOR RANK()**

**OUTPUT FOR DENSE\_RANK()**

****

**OUTPUT FOR ROW\_NUMBER()**

****

**Exercise 1: Create a Stored Procedure**

**SQL CODE:**

CREATE PROCEDURE ViewEmployees

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM Employees;

END;

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

**SQLCODE:**

CREATE PROCEDURE ViewEmployees

AS

BEGIN

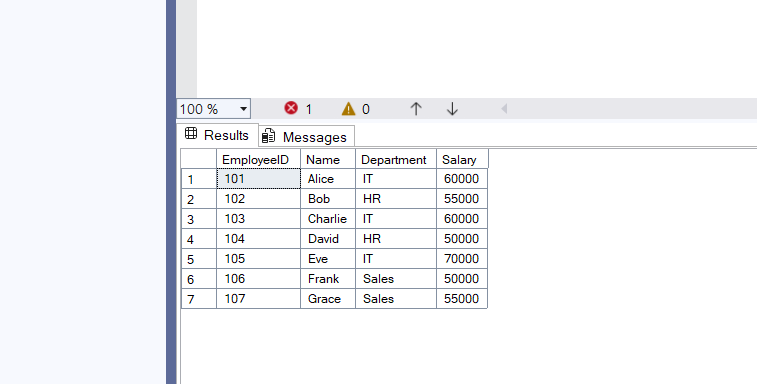
SET NOCOUNT ON;

SELECT \* FROM Employees;

END;

EXEC ViewEmployees;

**OUTPUT FOR RETURNING DATA:**

****