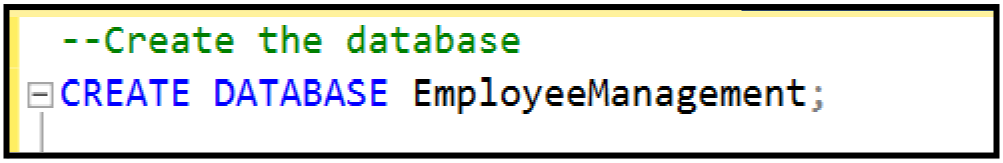
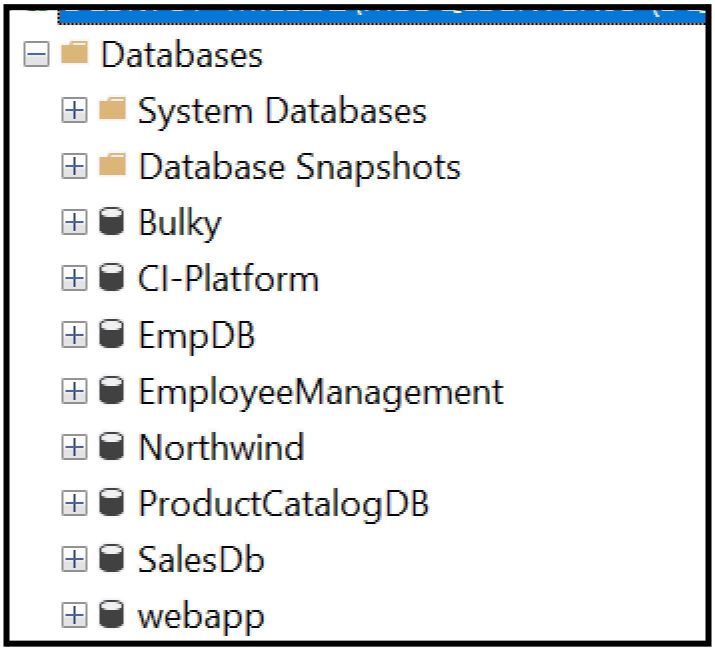
SQL Join Practical

Step 1: Create the Database

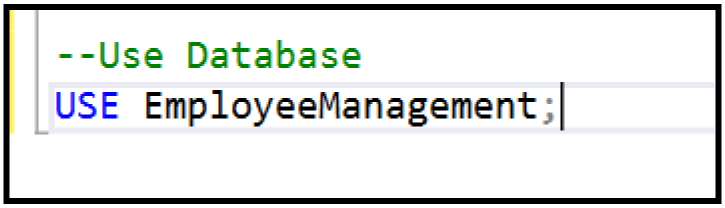


```sql
--Create the database
CREATE DATABASE EmployeeManagement;
```

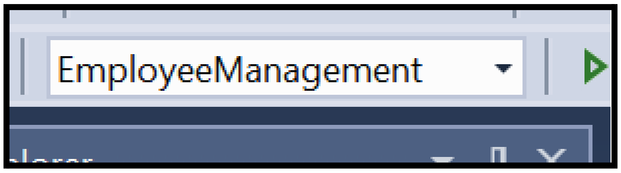


- Databases
  - System Databases
  - Database Snapshots
  - Bulky
  - CI-Platform
  - EmpDB
  - EmployeeManagement
  - Northwind
  - ProductCatalogDB
  - SalesDb
  - webapp

Step 2: Use the EmployeeManagement Database

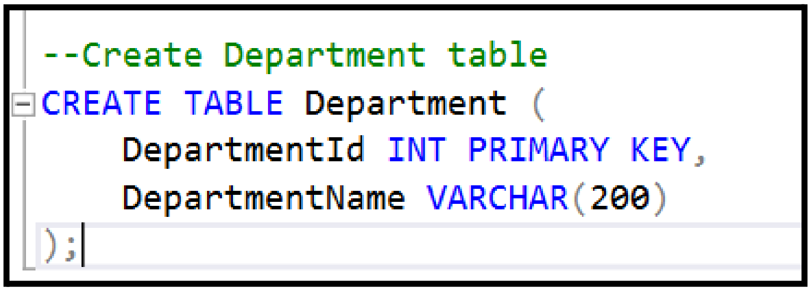


```sql
--Use Database
USE EmployeeManagement;
```

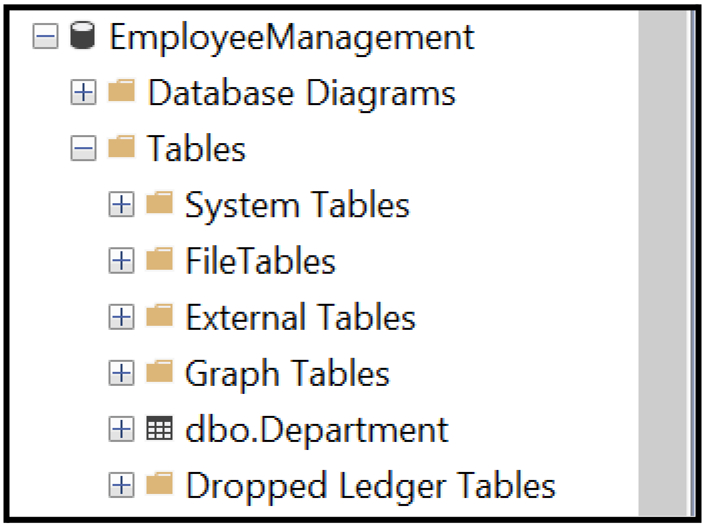


EmployeeManagement

Step 3: Create the Department table

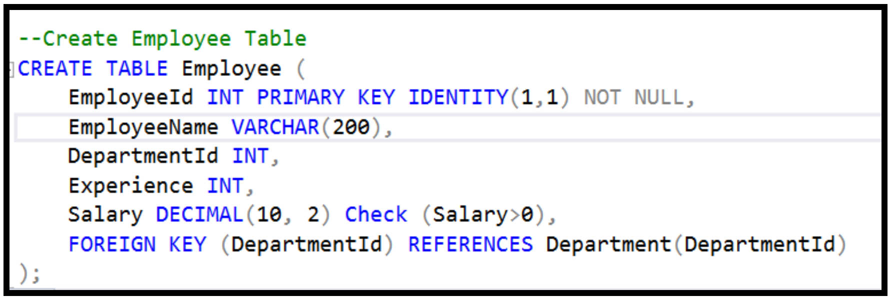


```
--Create Department table
CREATE TABLE Department (
    DepartmentId INT PRIMARY KEY,
    DepartmentName VARCHAR(200)
);
```

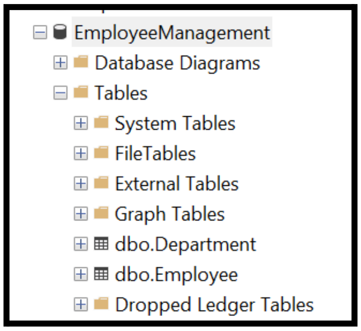


```
⊟ EmployeeManagement
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.Department
        ⊞ Dropped Ledger Tables
```

Step 4: Create Employee Table

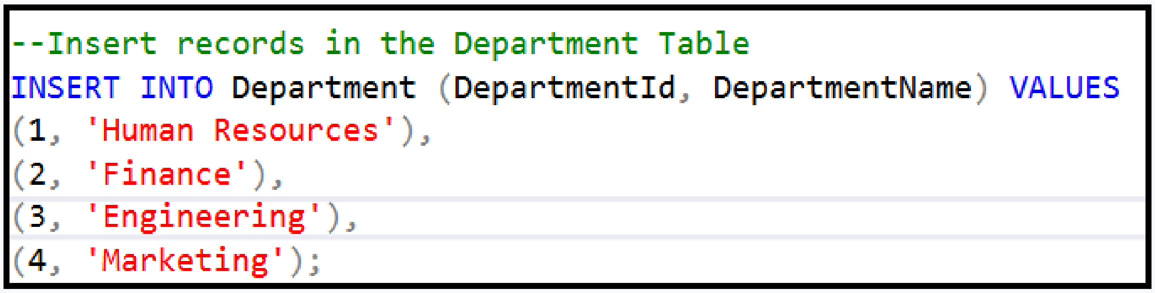


```
--Create Employee Table
CREATE TABLE Employee (
    EmployeeId INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    EmployeeName VARCHAR(200),
    DepartmentId INT,
    Experience INT,
    Salary DECIMAL(10, 2) Check (Salary>0),
    FOREIGN KEY (DepartmentId) REFERENCES Department(DepartmentId)
);
```

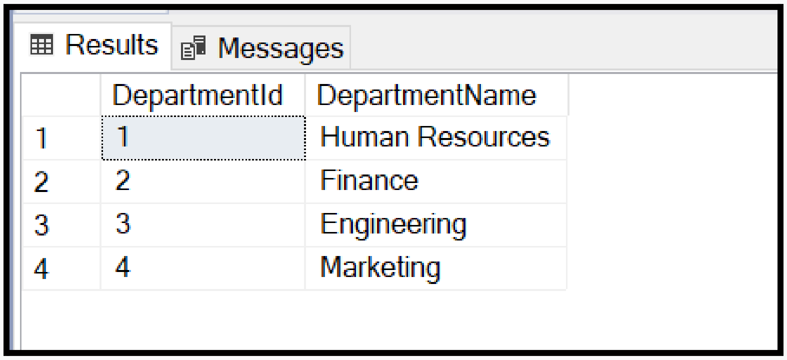


```
⊟ EmployeeManagement
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.Department
        ⊞ dbo.Employee
        ⊞ Dropped Ledger Tables
```

Step 5: Insert the records in the Department Table

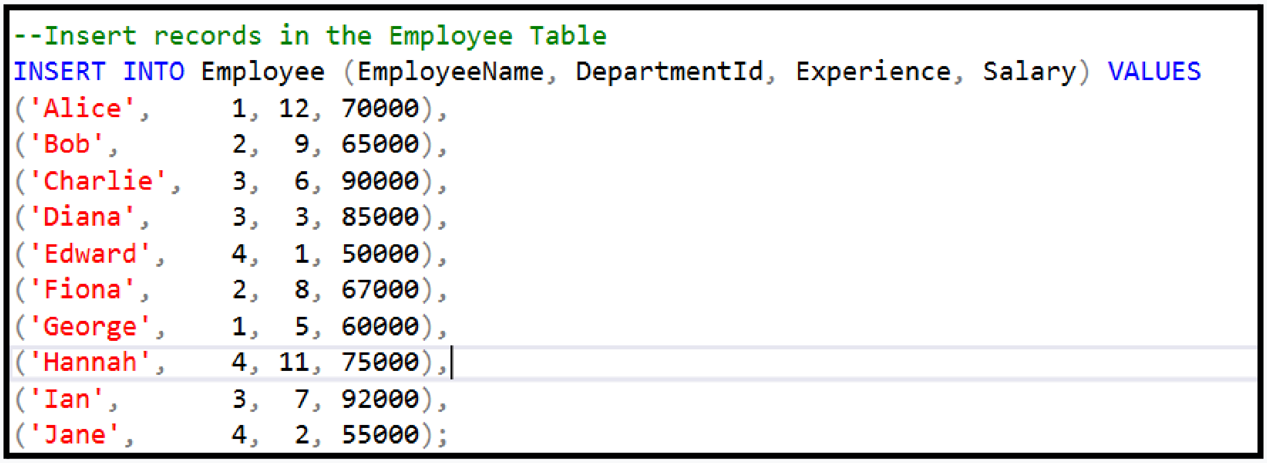


```sql
--Insert records in the Department Table
INSERT INTO Department (DepartmentId, DepartmentName) VALUES
(1, 'Human Resources'),
(2, 'Finance'),
(3, 'Engineering'),
(4, 'Marketing');
```

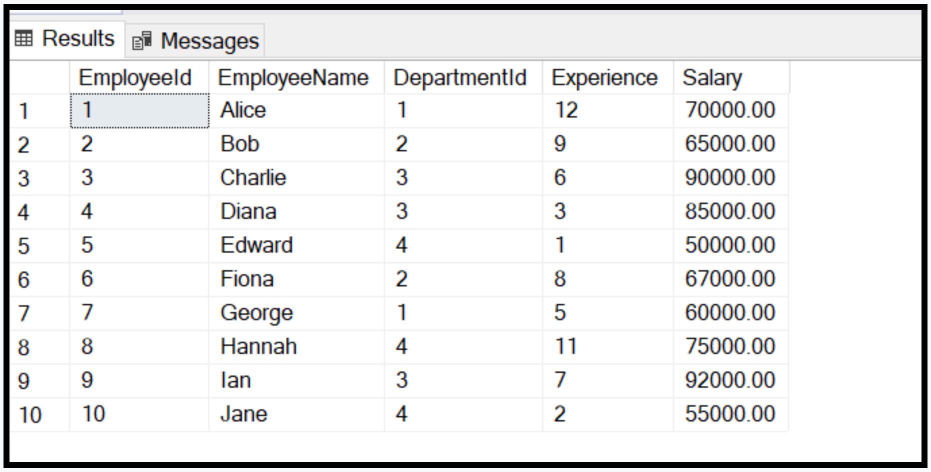


	DepartmentId	DepartmentName
1	1	Human Resources
2	2	Finance
3	3	Engineering
4	4	Marketing

Step 6: Insert records in the Employee Table

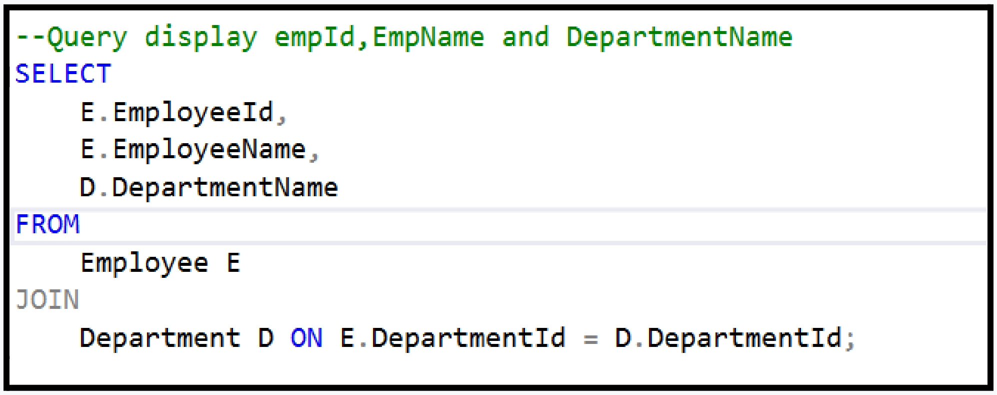


```sql
--Insert records in the Employee Table
INSERT INTO Employee (EmployeeName, DepartmentId, Experience, Salary) VALUES
('Alice',      1, 12, 70000),
('Bob',        2,  9, 65000),
('Charlie',    3,  6, 90000),
('Diana',      3,  3, 85000),
('Edward',     4,  1, 50000),
('Fiona',      2,  8, 67000),
('George',     1,  5, 60000),
('Hannah',     4, 11, 75000),
('Ian',        3,  7, 92000),
('Jane',       4,  2, 55000);
```

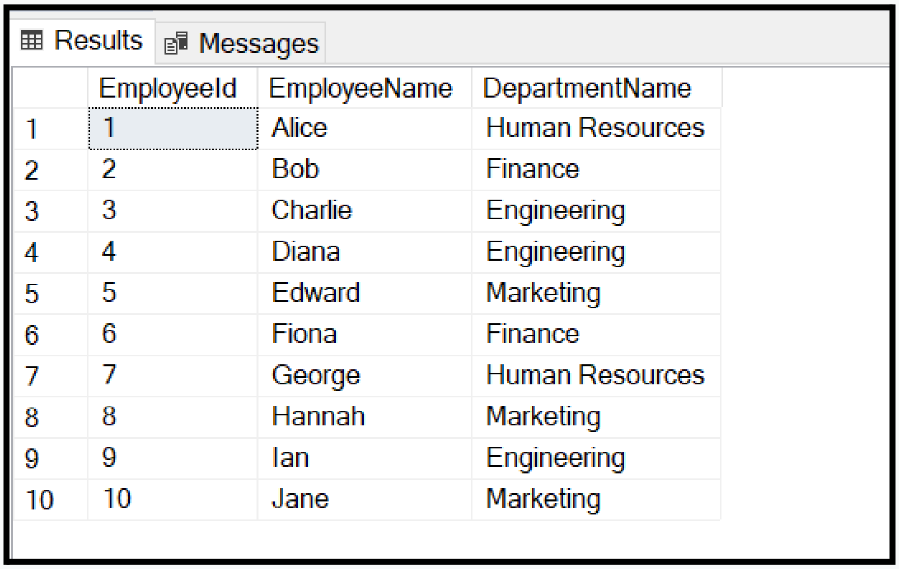


	EmployeeId	EmployeeName	DepartmentId	Experience	Salary
1	1	Alice	1	12	70000.00
2	2	Bob	2	9	65000.00
3	3	Charlie	3	6	90000.00
4	4	Diana	3	3	85000.00
5	5	Edward	4	1	50000.00
6	6	Fiona	2	8	67000.00
7	7	George	1	5	60000.00
8	8	Hannah	4	11	75000.00
9	9	Ian	3	7	92000.00
10	10	Jane	4	2	55000.00

Step 7: Query display empId,EmpName and DepartmentName

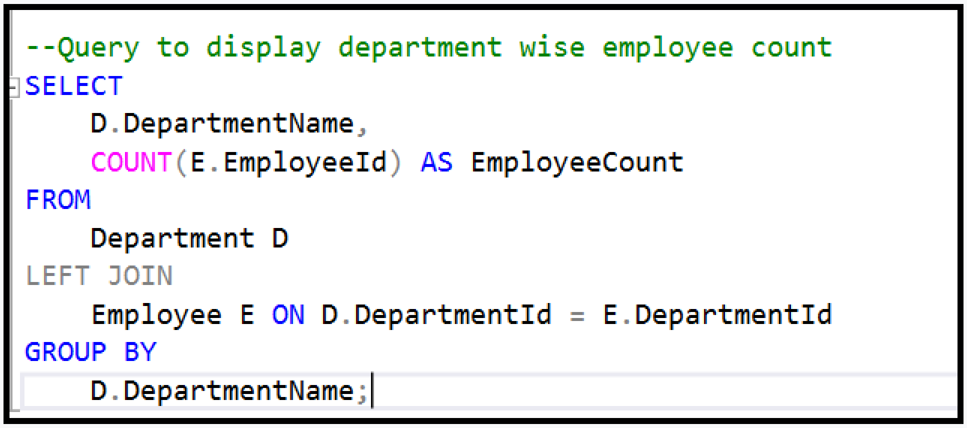


```
--Query display empId,EmpName and DepartmentName
SELECT
    E.EmployeeId,
    E.EmployeeName,
    D.DepartmentName
FROM
    Employee E
JOIN
    Department D ON E.DepartmentId = D.DepartmentId;
```

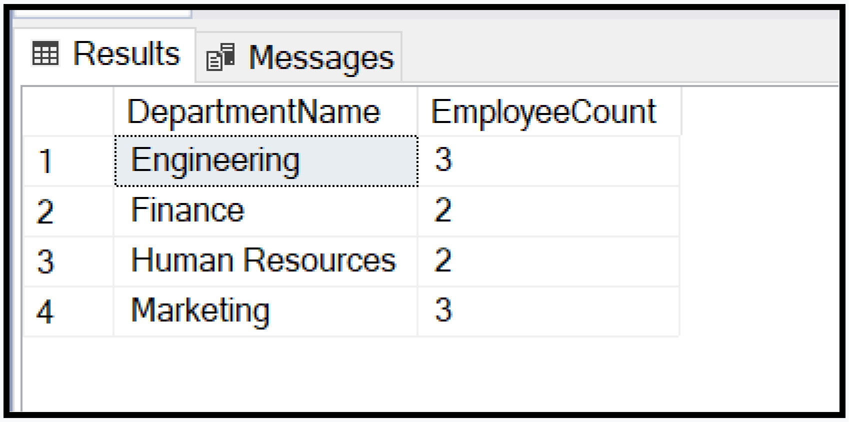


	EmployeeId	EmployeeName	DepartmentName
1	1	Alice	Human Resources
2	2	Bob	Finance
3	3	Charlie	Engineering
4	4	Diana	Engineering
5	5	Edward	Marketing
6	6	Fiona	Finance
7	7	George	Human Resources
8	8	Hannah	Marketing
9	9	Ian	Engineering
10	10	Jane	Marketing

Step 8: Query to display department wise employee count

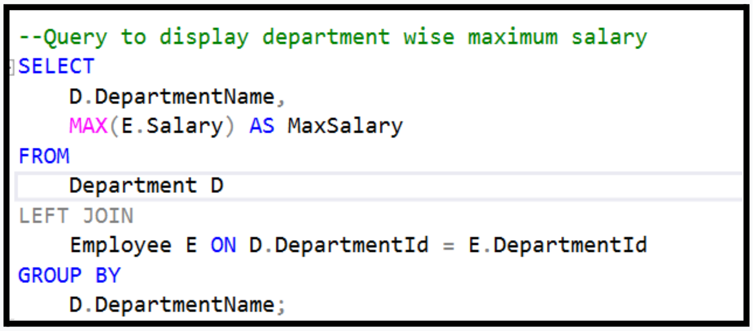


```sql
--Query to display department wise employee count
SELECT
    D.DepartmentName,
    COUNT(E.EmployeeId) AS EmployeeCount
FROM
    Department D
LEFT JOIN
    Employee E ON D.DepartmentId = E.DepartmentId
GROUP BY
    D.DepartmentName;
```

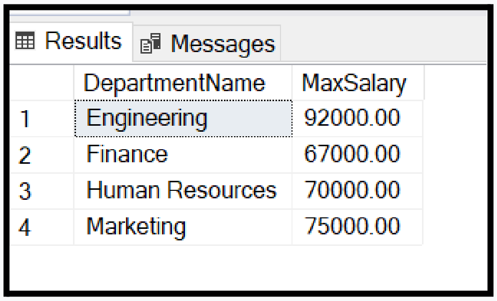


	DepartmentName	EmployeeCount
1	Engineering	3
2	Finance	2
3	Human Resources	2
4	Marketing	3

Step 9: Query to display department wise maximum salary

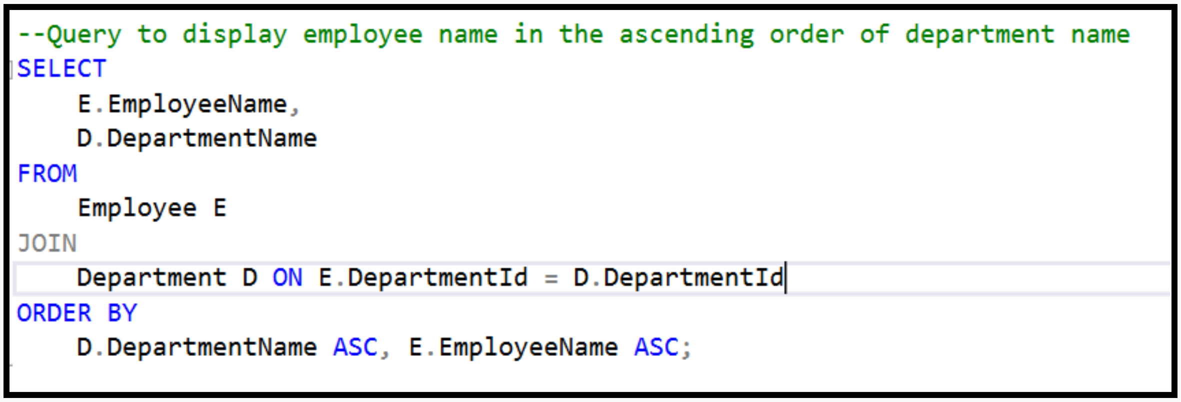


```sql
--Query to display department wise maximum salary
SELECT
    D.DepartmentName,
    MAX(E.Salary) AS MaxSalary
FROM
    Department D
LEFT JOIN
    Employee E ON D.DepartmentId = E.DepartmentId
GROUP BY
    D.DepartmentName;
```

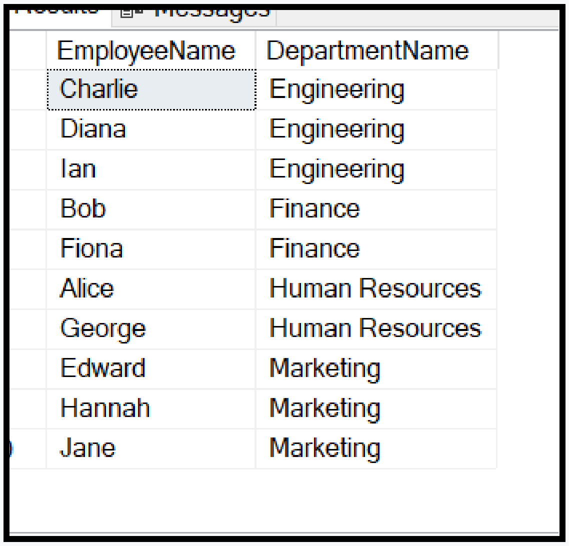


	DepartmentName	MaxSalary
1	Engineering	92000.00
2	Finance	67000.00
3	Human Resources	70000.00
4	Marketing	75000.00

Step 10: Query to display employee name in the ascending order of department name



```
--Query to display employee name in the ascending order of department name
SELECT
    E.EmployeeName,
    D.DepartmentName
FROM
    Employee E
JOIN
    Department D ON E.DepartmentId = D.DepartmentId
ORDER BY
    D.DepartmentName ASC, E.EmployeeName ASC;
```

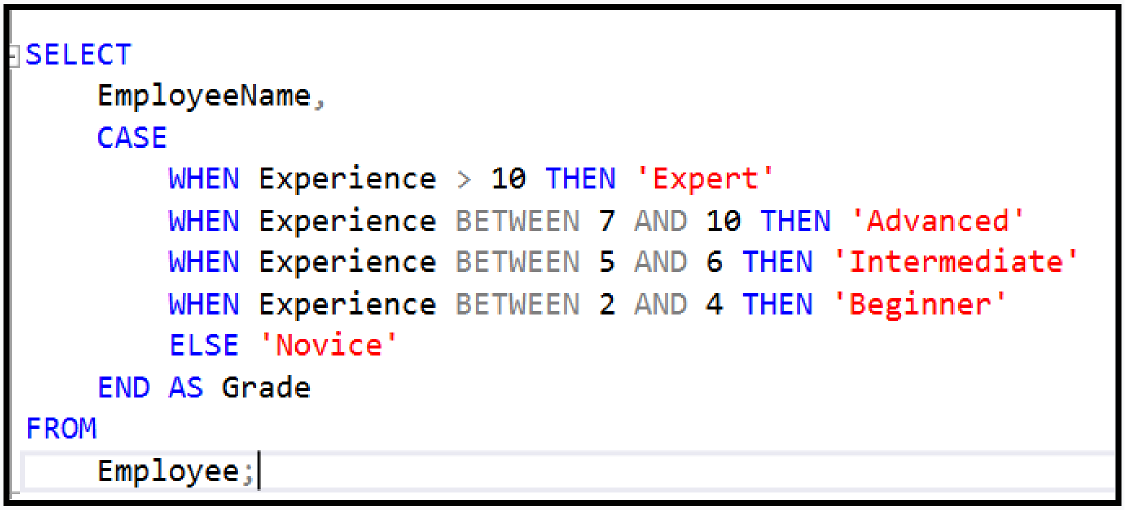


Results Messages

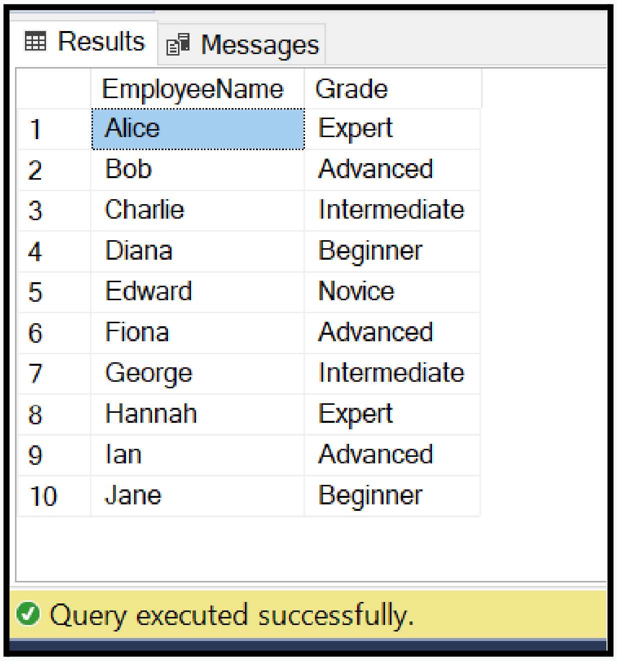
EmployeeName	DepartmentName
Charlie	Engineering
Diana	Engineering
Ian	Engineering
Bob	Finance
Fiona	Finance
Alice	Human Resources
George	Human Resources
Edward	Marketing
Hannah	Marketing
Jane	Marketing

Step:10 Query to display employeeName and grade where grade criteria is as below

A. **if employee has more than 10 years of experience then grade is expert**
B. **if employee has experience between 7 to 10 years then its grade is advanced**
C. **if employee has experience between 5 to 7 years then its grade is intermediate**
D. **if employee has experience between 2 to 5 year then its grade is beginner**
E. **and for the rest of case display grade as novice**



```sql
SELECT
    EmployeeName,
    CASE
        WHEN Experience > 10 THEN 'Expert'
        WHEN Experience BETWEEN 7 AND 10 THEN 'Advanced'
        WHEN Experience BETWEEN 5 AND 6 THEN 'Intermediate'
        WHEN Experience BETWEEN 2 AND 4 THEN 'Beginner'
        ELSE 'Novice'
    END AS Grade
FROM
    Employee;
```



	EmployeeName	Grade
1	Alice	Expert
2	Bob	Advanced
3	Charlie	Intermediate
4	Diana	Beginner
5	Edward	Novice
6	Fiona	Advanced
7	George	Intermediate
8	Hannah	Expert
9	Ian	Advanced
10	Jane	Beginner

✓ Query executed successfully.