**SQL Coding Challenge - 1**

**Name : Podutur Lahari**

**Date : 08-11-2024**

**Querying Data by Using Joins and Subqueries & subtotal**

**Joins :** A JOIN operation is used to combine records from two or more tables in a database. The tables are typically related by a common column (usually a foreign key). There are several types of joins, each with its specific behavior. Joins allow you to retrieve related data from multiple tables in a single query.

**Types of Joins:**

* INNER JOIN
* LEFT JOIN (or LEFT OUTER JOIN):
* RIGHT JOIN (or RIGHT OUTER JOIN):
* FULL JOIN (or FULL OUTER JOIN):
* SELF JOIN

**1. Employees who have the same department as another employee and their salary difference.(Self join)**

**Code :**

–Create Database

CREATE DATABASE day2db;

-- Create Employees table

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

Age INT,

Department NVARCHAR(50),

Salary DECIMAL(10, 2)

);

-- Insert sample data

INSERT INTO Employees (EmployeeID, FirstName, LastName, Age, Department, Salary) VALUES

(1, 'Arjun', 'Reddy', 30, 'HR', 55000),

(2, 'Manoj', 'Nair', 25, 'IT', 60000),

(3, 'Lakshmi', 'Iyer', 40, 'Finance', 70000),

(4, 'Venkatesh', 'Raj', 35, 'IT', 65000),

(5, 'Sneha', 'Menon', 28, 'HR', 58000);

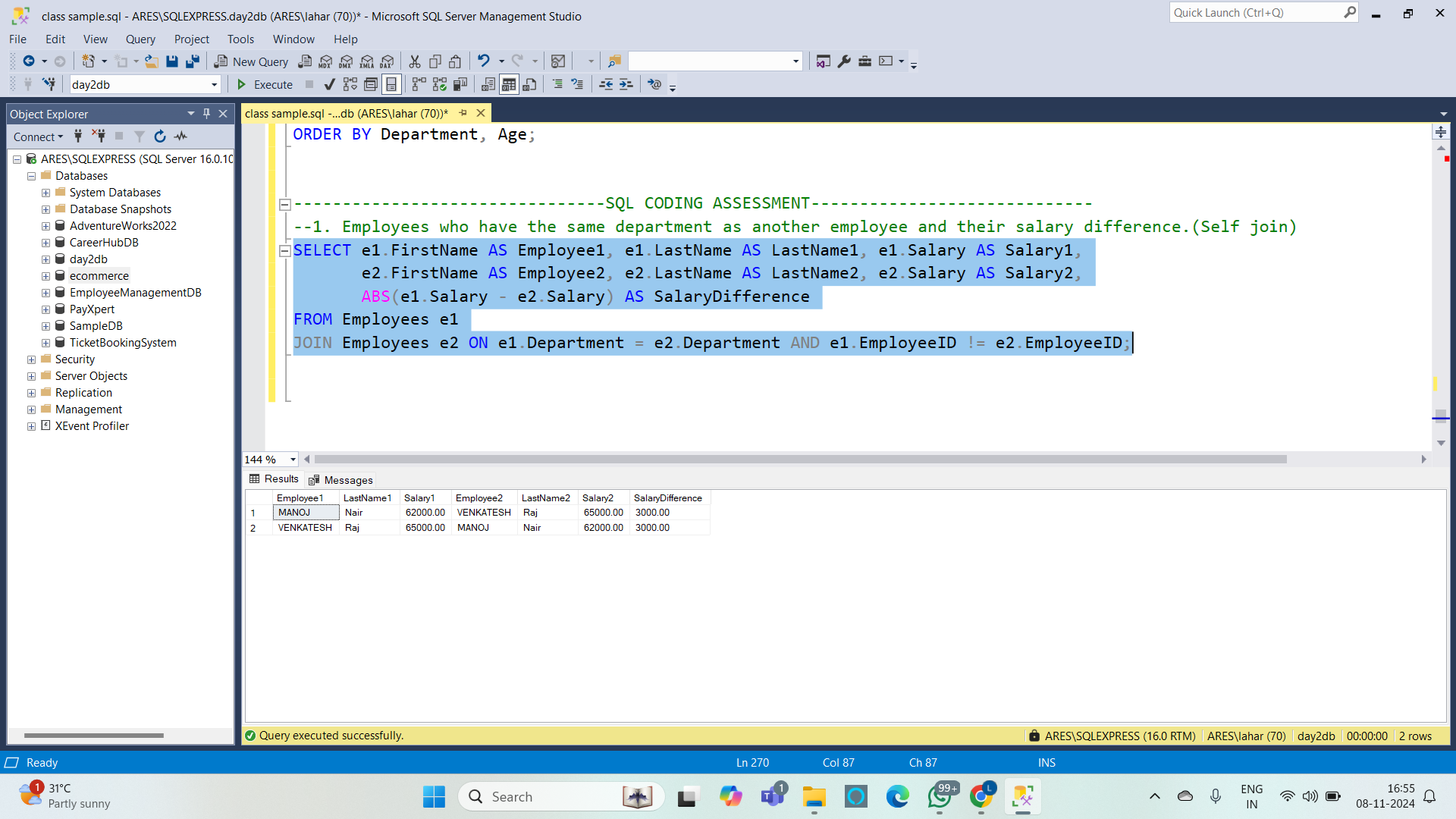
SELECT e1.FirstName AS Employee1, e1.LastName AS LastName1, e1.Salary AS Salary1,

e2.FirstName AS Employee2, e2.LastName AS LastName2, e2.Salary AS Salary2,

ABS(e1.Salary - e2.Salary) AS SalaryDifference

FROM Employees e1

JOIN Employees e2 ON e1.Department = e2.Department AND e1.EmployeeID != e2.EmployeeID;



**Subqueries:** A subquery (or inner query) is a query embedded within another query. It is used to retrieve intermediate results that are then used in the main query. Subqueries can be used in various SQL clauses, such as SELECT, FROM, WHERE, or HAVING.

#### **Types of Subqueries:**

* **Scalar Subquery**: Returns a single value (used in SELECT or WHERE clauses).
* **Inline Subquery**: Used in the FROM clause to provide a derived table.
* **Correlated Subquery**: Depends on values from the outer query, typically used in the WHERE clause.

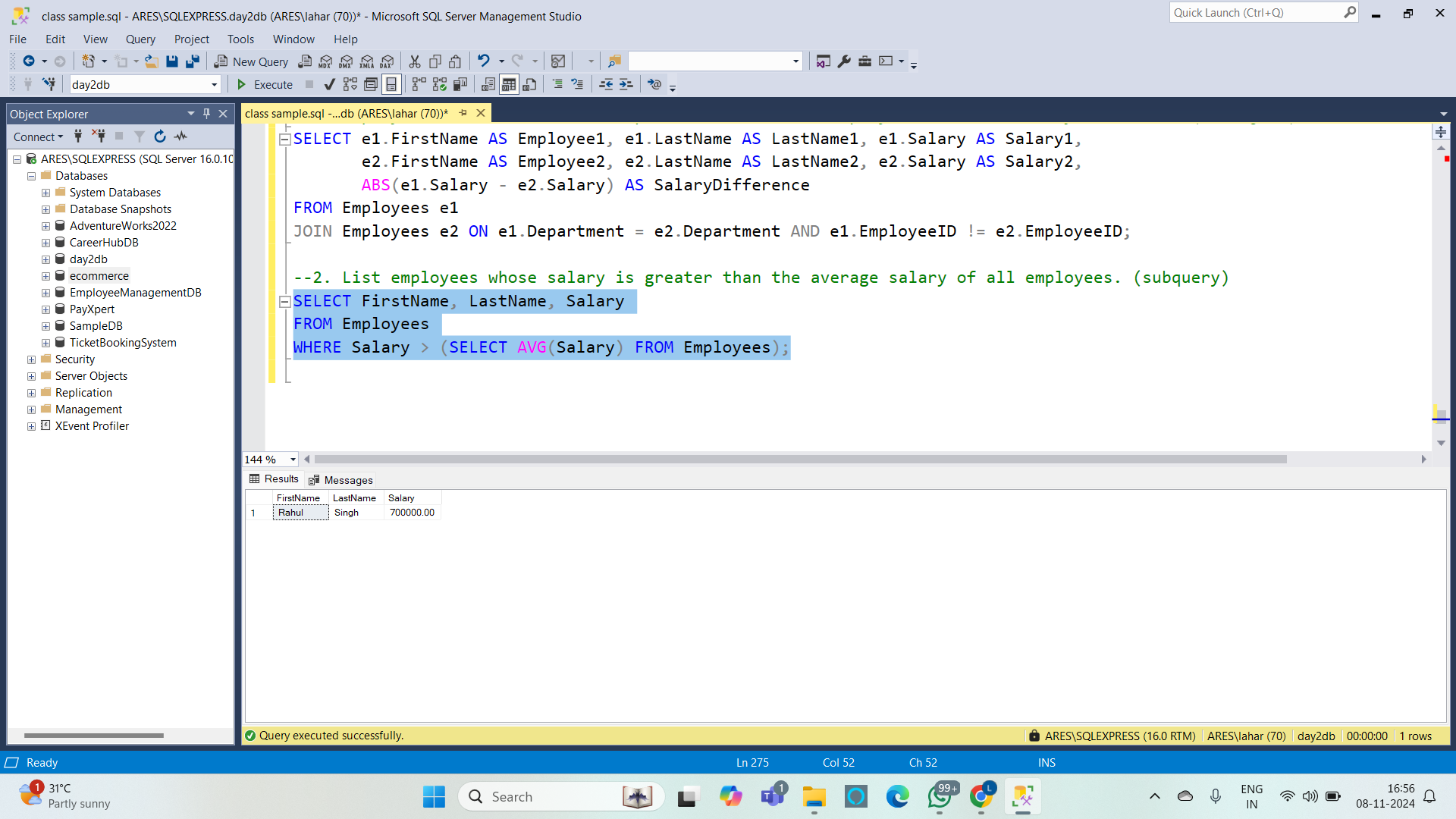
**2. List employees whose salary is greater than the average salary of all employees. (subquery)**

**Code :**

SELECT FirstName, LastName, Salary

FROM Employees

WHERE Salary > (SELECT AVG(Salary) FROM Employees);



**3. Get the average salary for each department (Query Using a Subquery in the FROM Clause)**

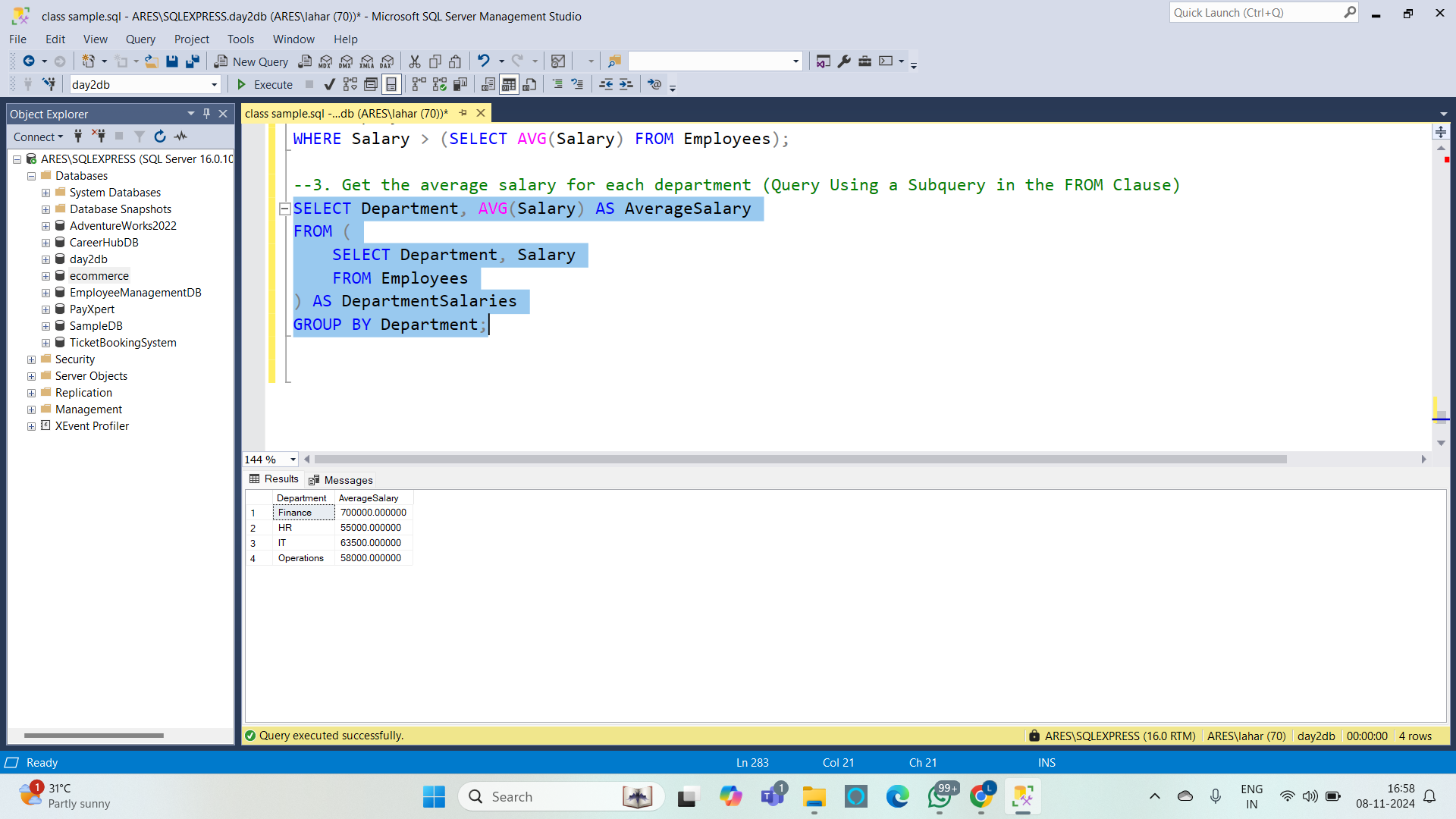
**Code :**

SELECT Department, AVG(Salary) AS AverageSalary

FROM (SELECT Department, SalaryFROM Employees)

AS DepartmentSalaries

GROUP BY Department;



**Subtotals :** Subtotals are used to calculate intermediate aggregation results, typically in reporting scenarios. The GROUP BY clause is used to group data by one or more columns, and then aggregate functions like SUM(), AVG(), COUNT(), etc., are applied to each group.

A subtotal can be calculated by grouping data, often followed by adding a final row for the grand total using a UNION ALL.

**4. Show each department and the total salary of employees in that department, along with a subtotal row indicating the total salary across all departments (Query Using Subtotal with GROUP BY)**

**Code :**

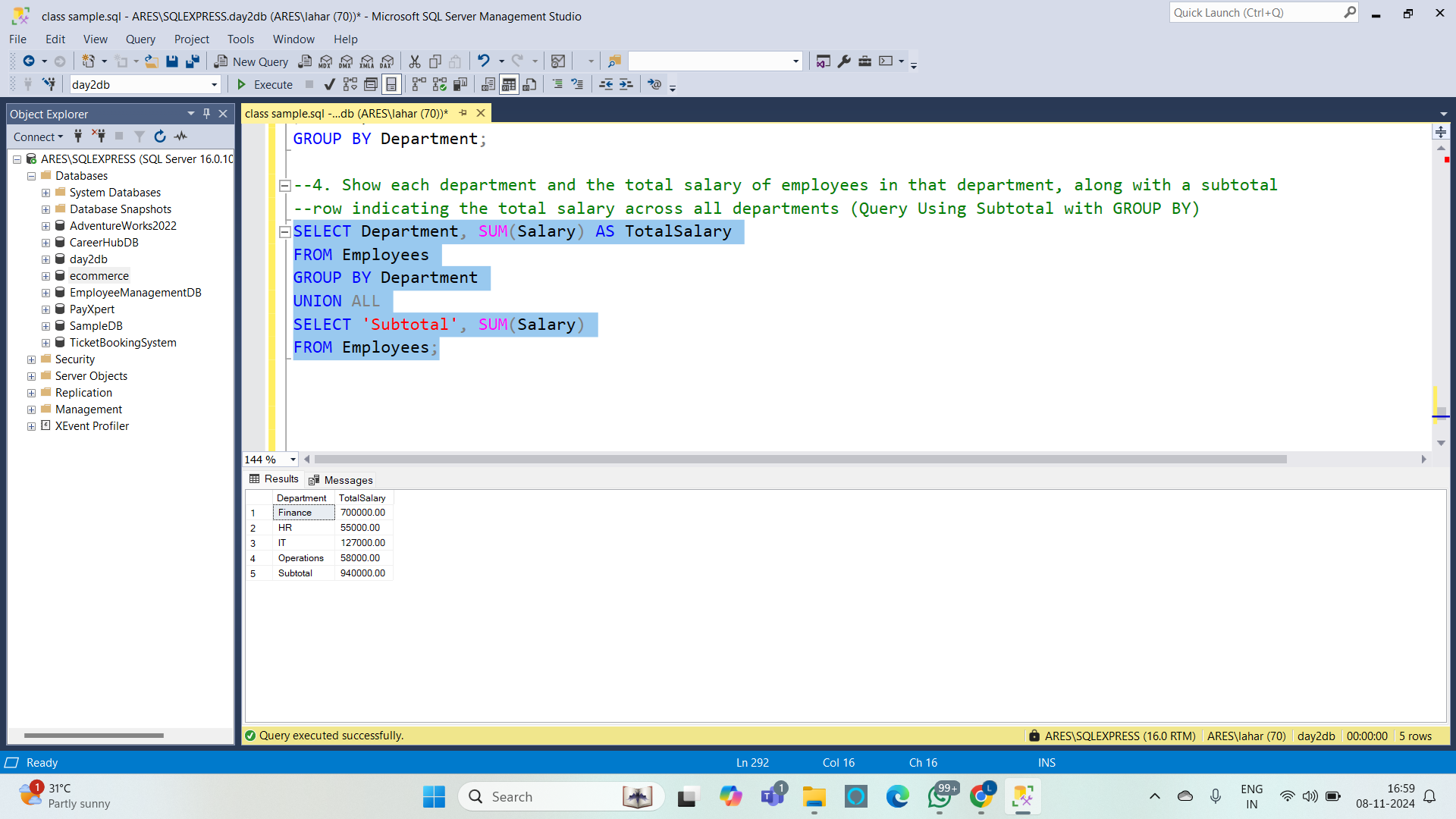
SELECT Department, SUM(Salary) AS TotalSalary

FROM Employees

GROUP BY Department

UNION ALL

SELECT 'Subtotal', SUM(Salary)

FROM Employees;

**5.List each department's average salary and the employees who earn more than the average salary in their respective departments, with a subtotal for each department's salary total(combines self-joins, subqueries, and subtotals)**

**Code :**

SELECT e.Department, e.FirstName, e.LastName, e.Salary, d.AvgSalary

FROM Employees e

JOIN (

SELECT Department, AVG(Salary) AS AvgSalary

FROM Employees

GROUP BY Department

) d ON e.Department = d.Department

WHERE e.Salary > d.AvgSalary

UNION ALL

SELECT Department, 'Subtotal' AS FirstName, NULL AS LastName, SUM(Salary) AS Salary, NULL AS AvgSalary

FROM Employees

GROUP BY Department

ORDER BY Department, FirstName;

