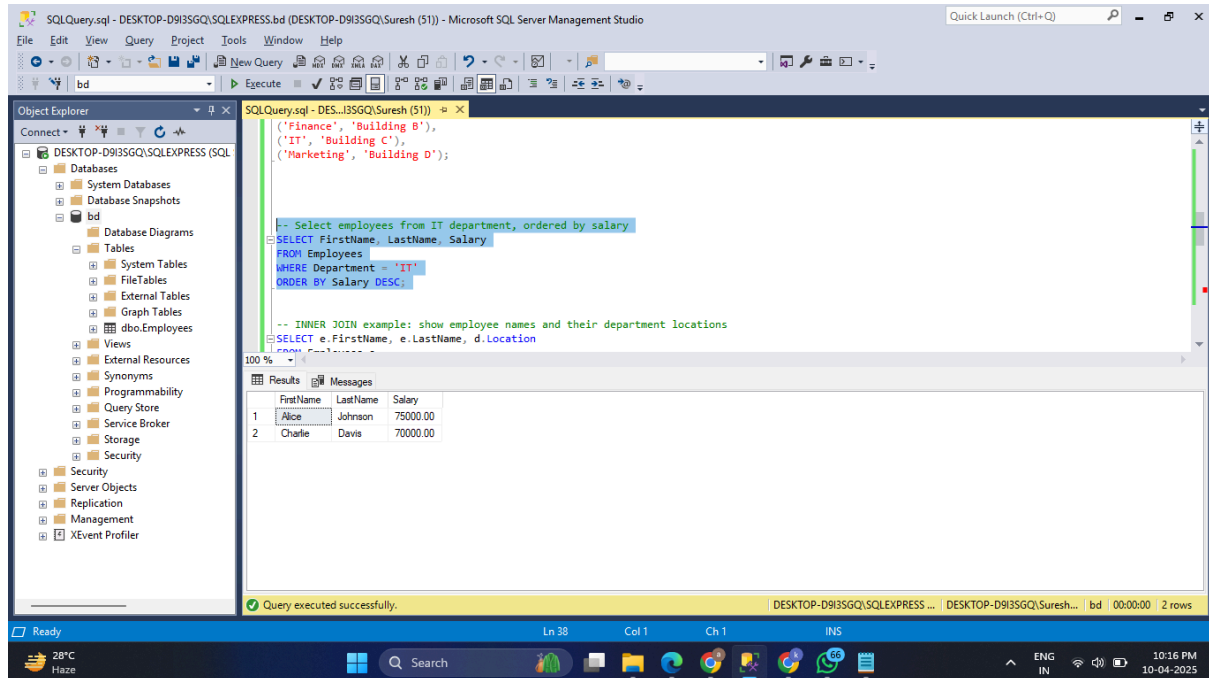
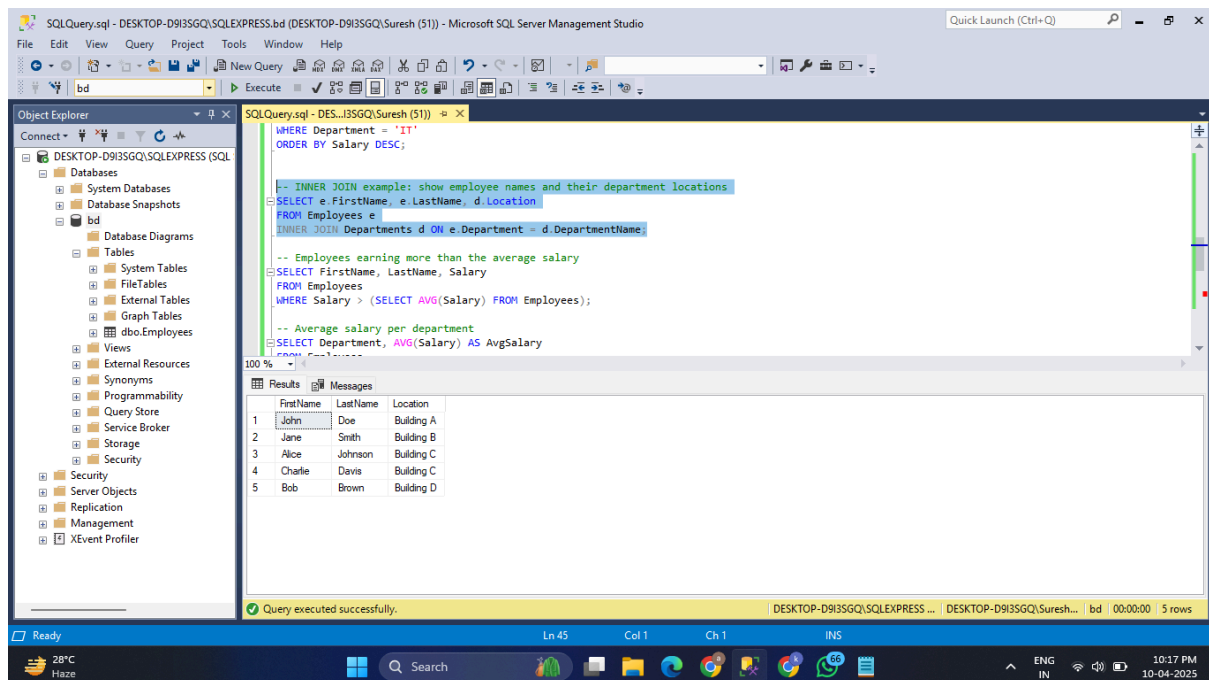


-- Select employees from IT department, ordered by salary  
 SELECT FirstName, LastName, Salary  
 FROM Employees  
 WHERE Department = 'IT'  
 ORDER BY Salary DESC;



Q2

-- INNER JOIN example: show employee names and their department locations  
 SELECT e.FirstName, e.LastName, d.Location  
 FROM Employees e  
 INNER JOIN Departments d ON e.Department = d.DepartmentName;



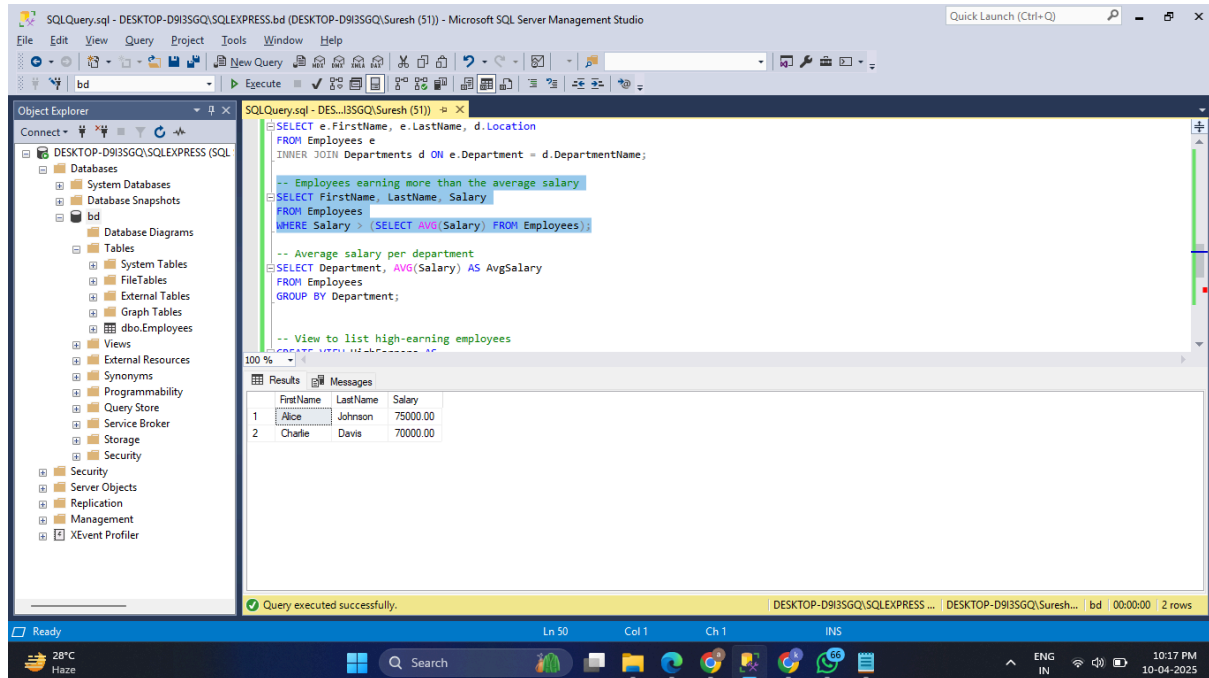
Q3.

-- Employees earning more than the average salary

SELECT FirstName, LastName, Salary

FROM Employees

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



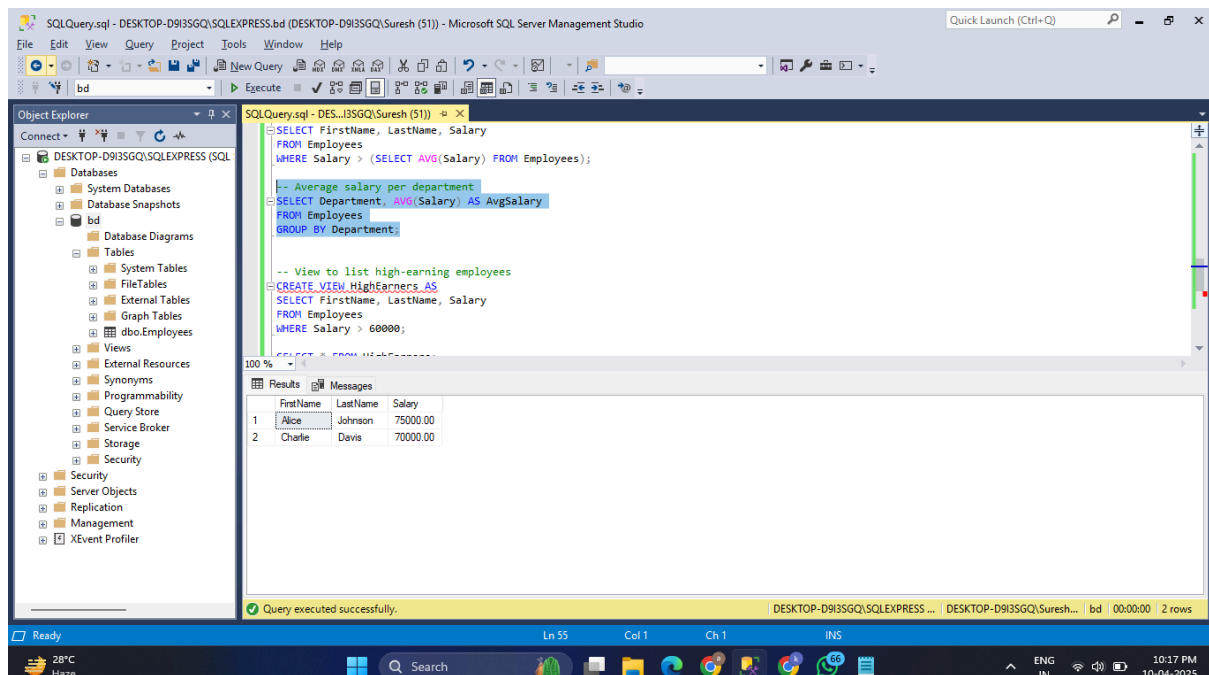
Q4.

-- Average salary per department

SELECT Department, AVG(Salary) AS AvgSalary

FROM Employees

GROUP BY Department;



Q5.

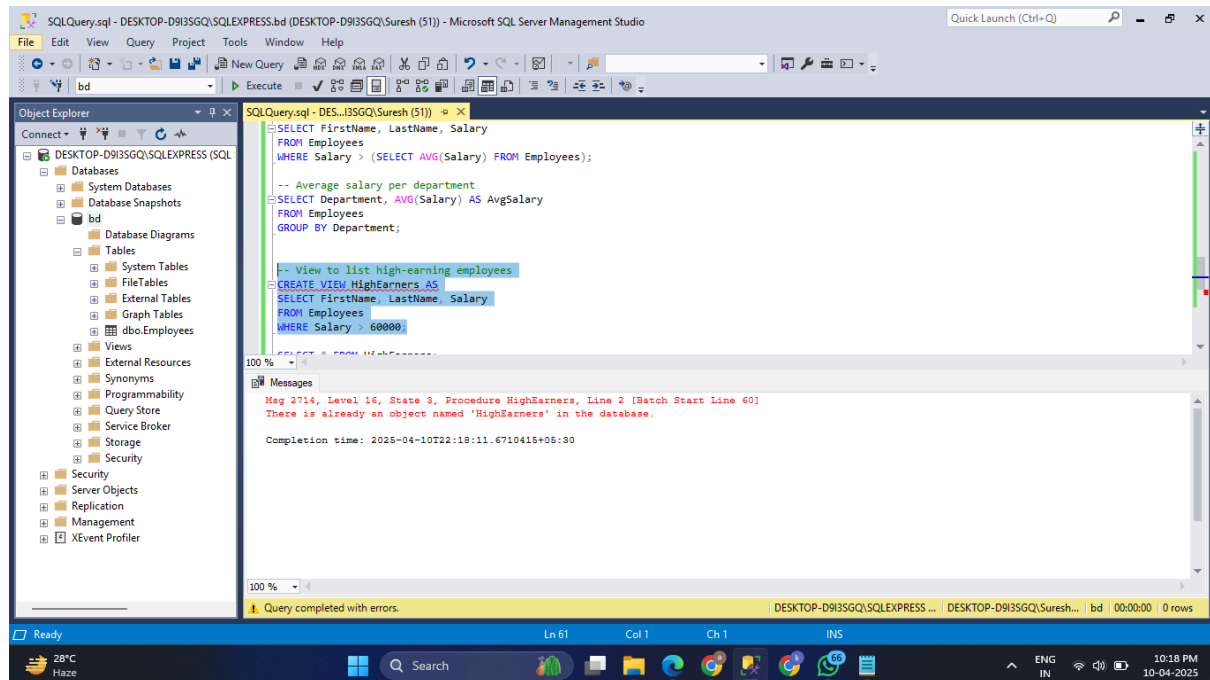
-- View to list high-earning employees

CREATE VIEW HighEarnings AS

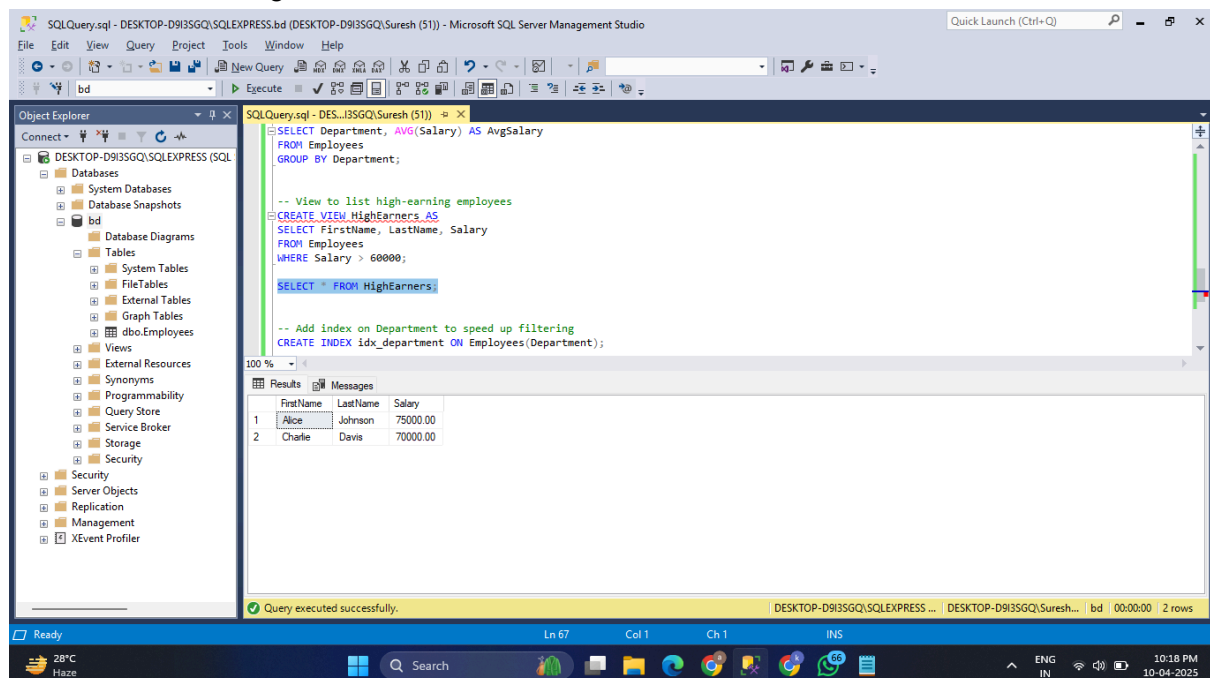
SELECT FirstName, LastName, Salary

FROM Employees

WHERE Salary > 60000;



SELECT \* FROM HighEarnings;



Q6.

-- Add index on Department to speed up filtering  
CREATE INDEX idx\_department ON Employees(Department);

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'bd' selected. The central query editor contains the following SQL script:

```
-- View to list high-earning employees
CREATE VIEW HighEarnings AS
SELECT FirstName, LastName, Salary
FROM Employees
WHERE Salary > 60000;

SELECT * FROM HighEarnings;

-- Add index on Department to speed up filtering
CREATE INDEX idx_department ON Employees(Department);
```

The right pane shows the Results tab with the following data:

	FirstName	LastName	Salary
1	Alice	Johnson	75000.00
2	Charlie	Davis	70000.00

The status bar at the bottom indicates 'Query executed successfully.' and '2 rows'.