Employee Dataset SQL Analysis on Azure by Rohit Raman

"CTE, Ranking Functions, and Aggregation in SQL: Employee Dataset Analysis"

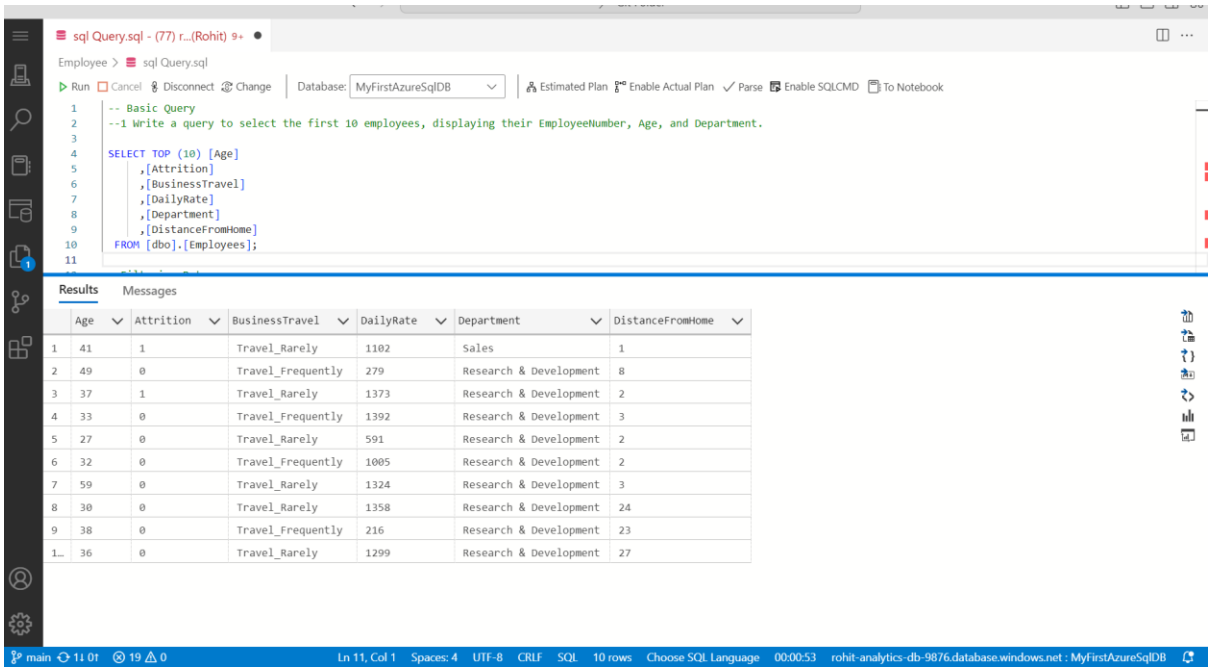
Introduction

This project focuses on analyzing employee data using SQL on Azure platform to extract insights on salary distribution, department composition, attrition rates, and employee rankings. The analysis is structured into fundamental SQL operations such as **basic queries, filtering, sorting, grouping, and aggregation**, along with more advanced **Common Table Expressions (CTEs) and ranking functions**.

Key analyses performed include retrieving specific employee details, filtering records based on conditions (e.g., employees older than 40 or those working in Sales), and sorting employees by salary. Aggregation techniques such as **COUNT (), AVG (), MIN (), and MAX ()** were used to measure attrition, calculate average department salaries, and determine the youngest and oldest employees per department.

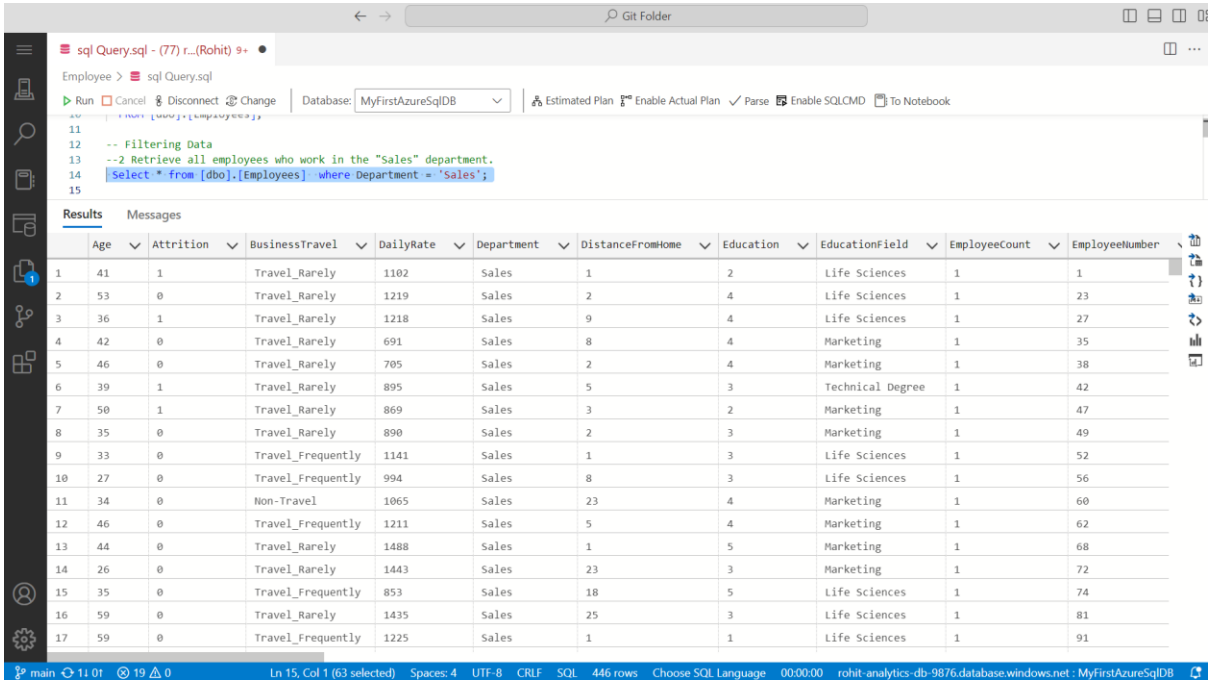
CTEs were utilized for more complex queries, including identifying employees with over **10 years of experience and above-average salaries**, ranking employees by income within departments, and calculating department-wise employee counts. Window functions like **RANK (), DENSE_RANK (), and ROW_NUMBER ()** were employed to rank employees based on income and experience, as well as to determine the second-highest salary and top earners per department. **PARTITION BY** was used to compare salaries within departments efficiently.

1. Basic Query: Select the first 10 employees, displaying their EmployeeNumber, Age, and Department.



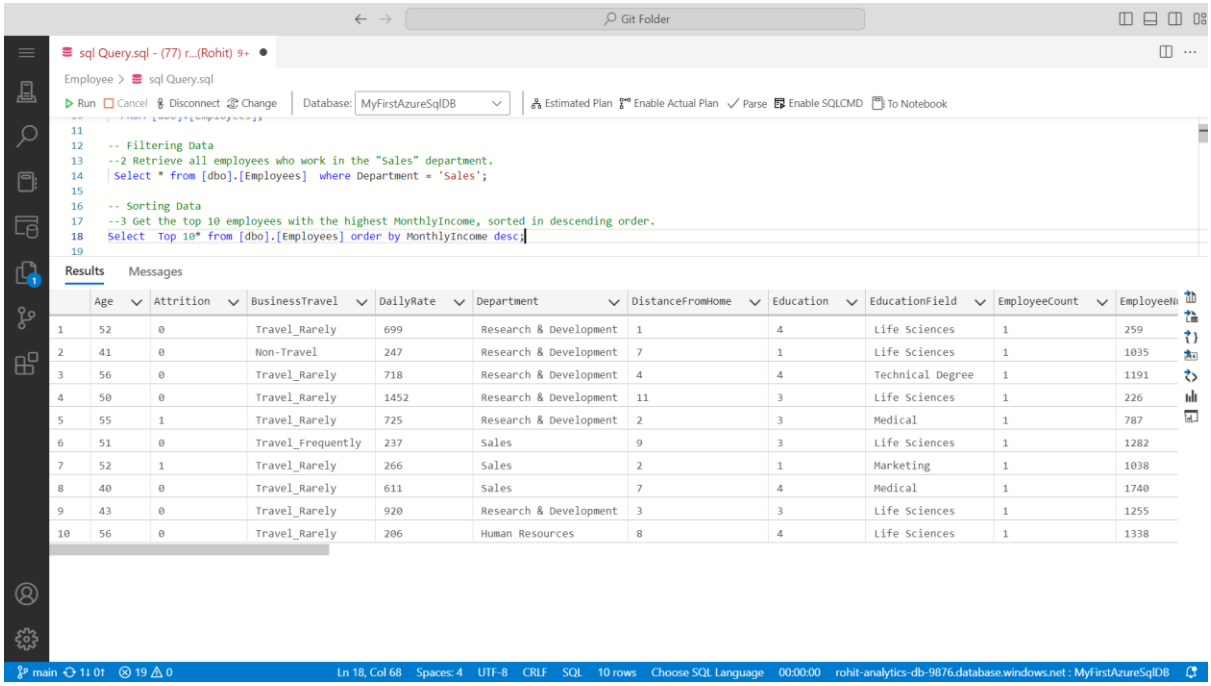
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome
1	41	1	Travel_Rarely	1102	Sales	1
2	49	0	Travel_Frequently	279	Research & Development	8
3	37	1	Travel_Rarely	1373	Research & Development	2
4	33	0	Travel_Frequently	1392	Research & Development	3
5	27	0	Travel_Rarely	591	Research & Development	2
6	32	0	Travel_Frequently	1005	Research & Development	2
7	59	0	Travel_Rarely	1324	Research & Development	3
8	30	0	Travel_Rarely	1358	Research & Development	24
9	38	0	Travel_Frequently	216	Research & Development	23
1...	36	0	Travel_Rarely	1299	Research & Development	27

2. Filtering Data: Retrieve all employees who work in the "Sales" department.



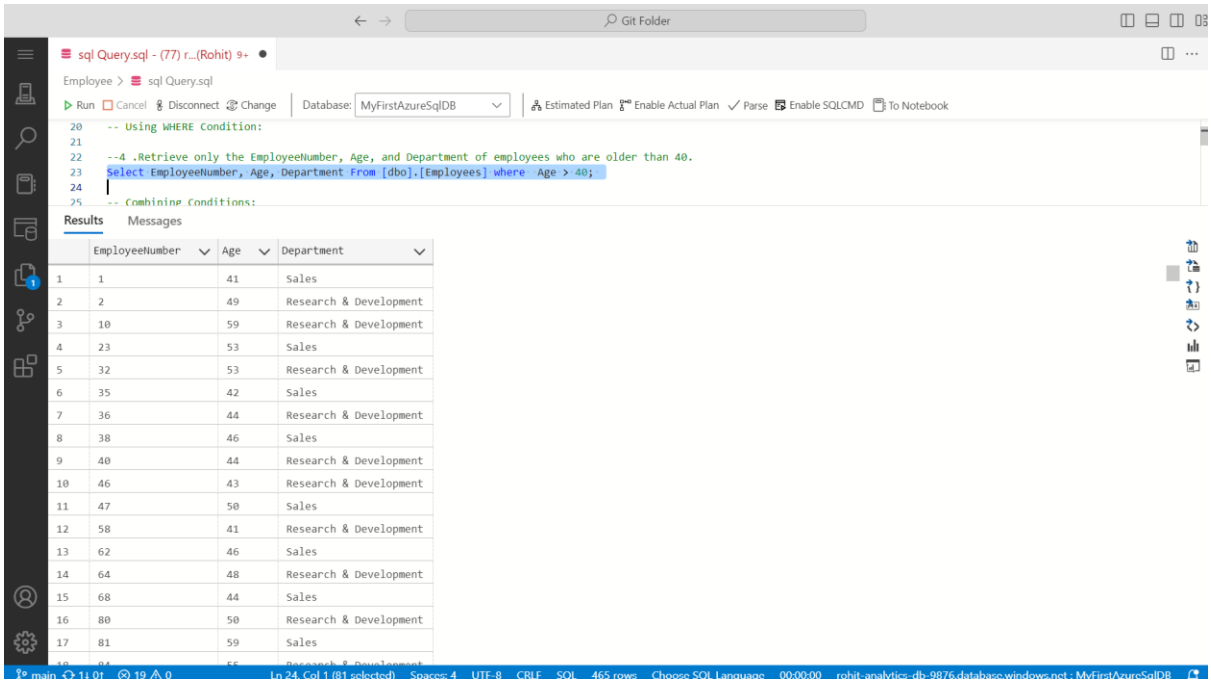
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
1	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
2	53	0	Travel_Rarely	1219	Sales	2	4	Life Sciences	1	23
3	36	1	Travel_Rarely	1218	Sales	9	4	Life Sciences	1	27
4	42	0	Travel_Rarely	691	Sales	8	4	Marketing	1	35
5	46	0	Travel_Rarely	705	Sales	2	4	Marketing	1	38
6	39	1	Travel_Rarely	895	Sales	5	3	Technical Degree	1	42
7	50	1	Travel_Rarely	869	Sales	3	2	Marketing	1	47
8	35	0	Travel_Rarely	890	Sales	2	3	Marketing	1	49
9	33	0	Travel_Frequently	1141	Sales	1	3	Life Sciences	1	52
10	27	0	Travel_Frequently	994	Sales	8	3	Life Sciences	1	56
11	34	0	Non-Travel	1065	Sales	23	4	Marketing	1	60
12	46	0	Travel_Frequently	1211	Sales	5	4	Marketing	1	62
13	44	0	Travel_Rarely	1488	Sales	1	5	Marketing	1	68
14	26	0	Travel_Rarely	1443	Sales	23	3	Marketing	1	72
15	35	0	Travel_Frequently	853	Sales	18	5	Life Sciences	1	74
16	59	0	Travel_Rarely	1435	Sales	25	3	Life Sciences	1	81
17	59	0	Travel_Frequently	1225	Sales	1	1	Life Sciences	1	91

3. Sorting Data: Get the top 10 employees with the highest MonthlyIncome, sorted in descending order.

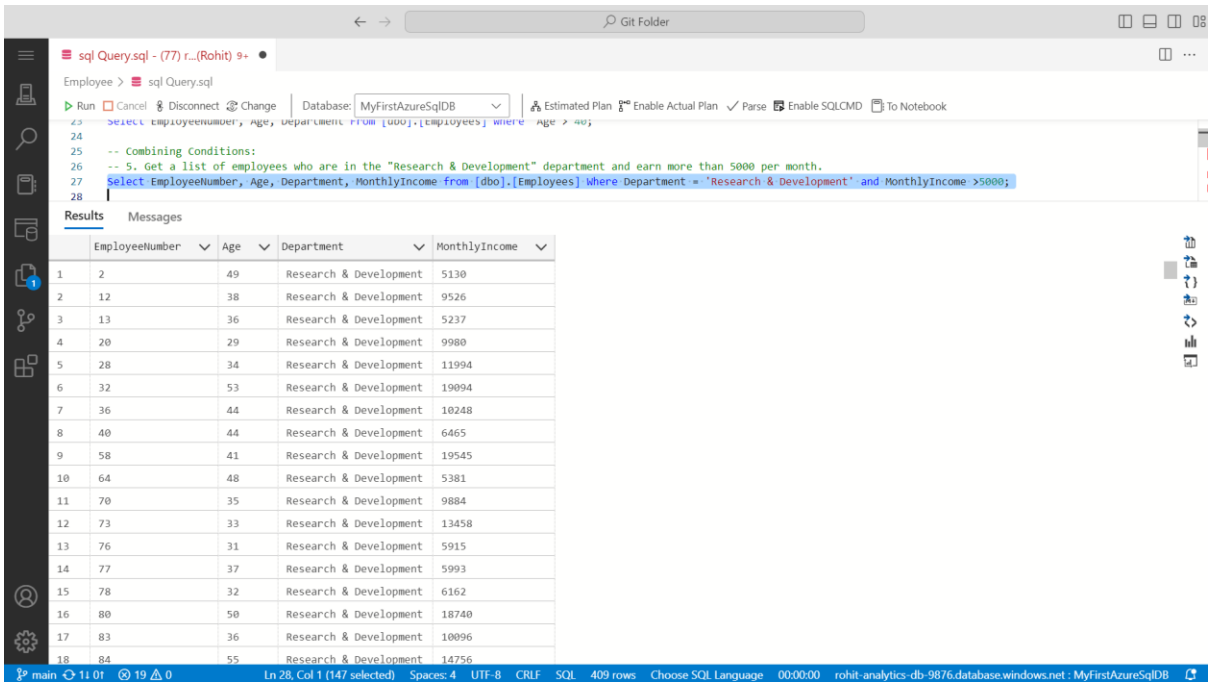


	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeN
1	52	0	Travel_Rarely	699	Research & Development	1	4	Life Sciences	1	259
2	41	0	Non-Travel	247	Research & Development	7	1	Life Sciences	1	1035
3	56	0	Travel_Rarely	718	Research & Development	4	4	Technical Degree	1	1191
4	50	0	Travel_Rarely	1452	Research & Development	11	3	Life Sciences	1	226
5	55	1	Travel_Rarely	725	Research & Development	2	3	Medical	1	787
6	51	0	Travel_Frequently	237	Sales	9	3	Life Sciences	1	1282
7	52	1	Travel_Rarely	266	Sales	2	1	Marketing	1	1038
8	40	0	Travel_Rarely	611	Sales	7	4	Medical	1	1740
9	43	0	Travel_Rarely	920	Research & Development	3	3	Life Sciences	1	1255
10	56	0	Travel_Rarely	206	Human Resources	8	4	Life Sciences	1	1338

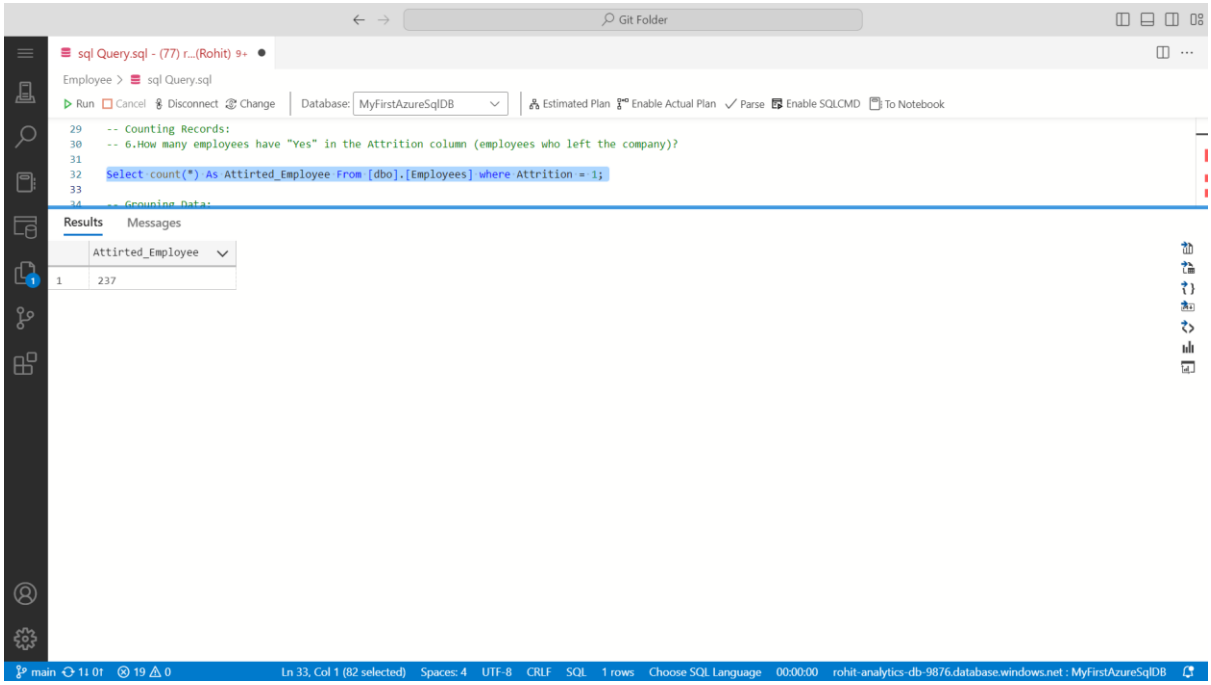
4. Using WHERE Condition: Retrieve EmployeeNumber, Age, and Department of employees older than 40.



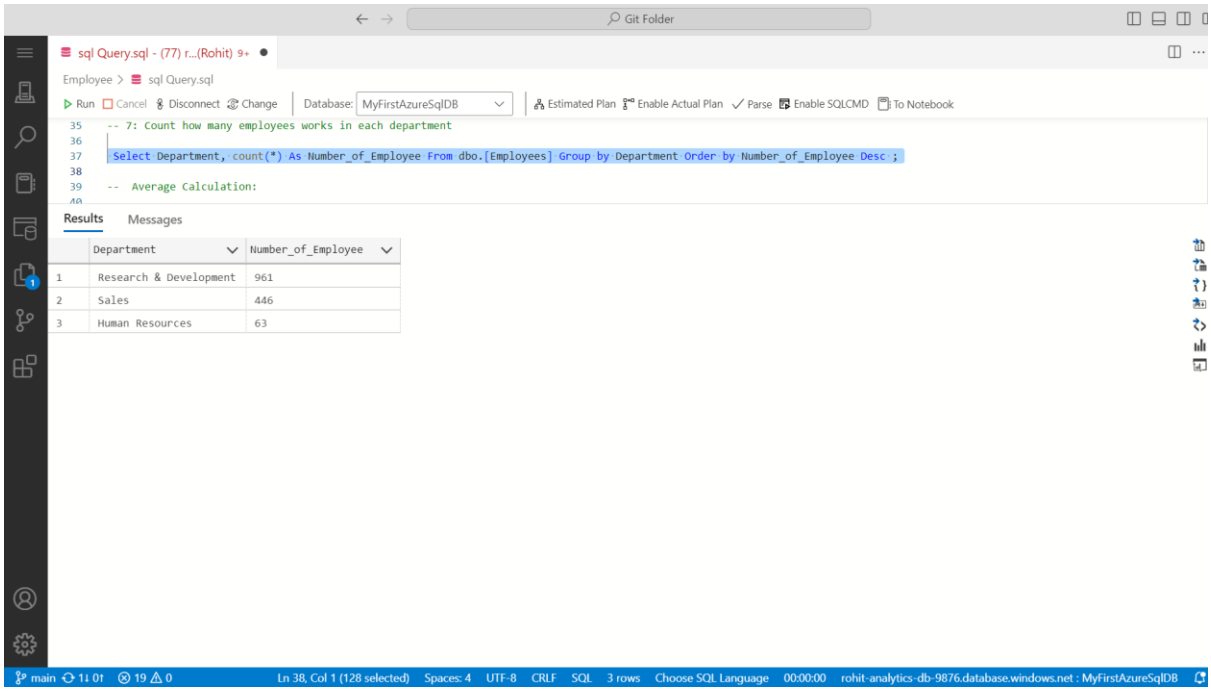
5. Combining Conditions: Get employees in "Research & Development" earning more than 5000 per month.



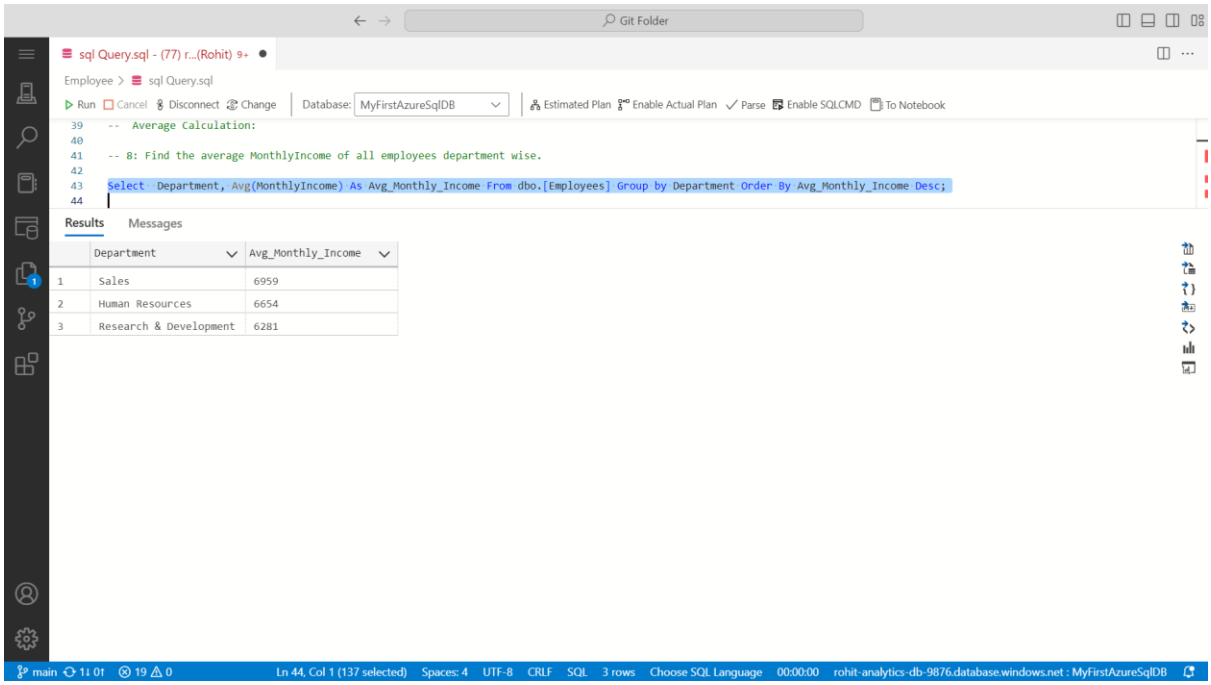
6. Counting Records: Count employees who left the company (Attrition = "Yes").



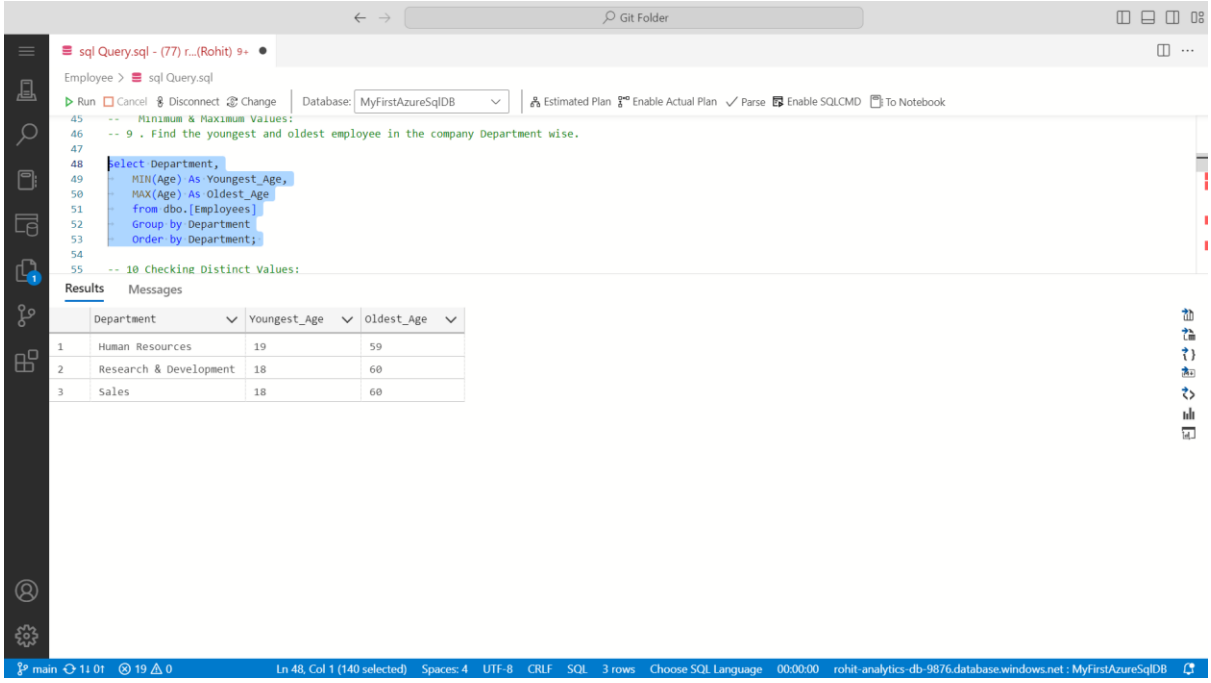
7. Grouping Data: Count employees in each department.



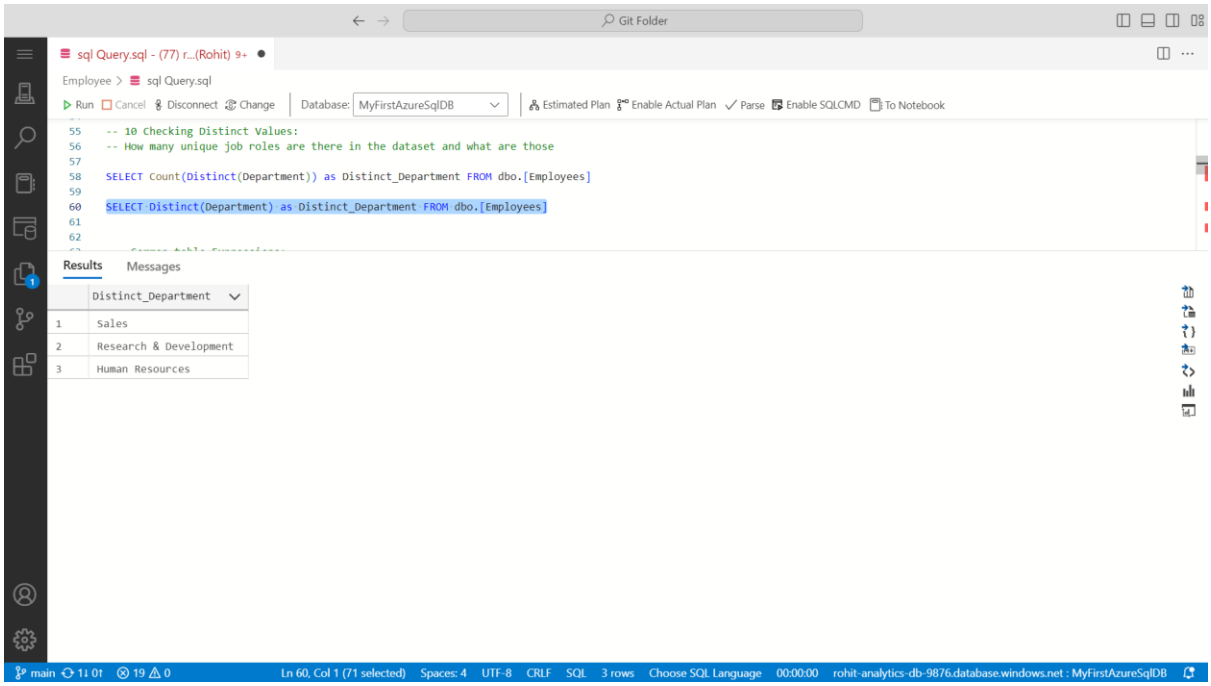
8. Average Calculation: Find the average MonthlyIncome by department.



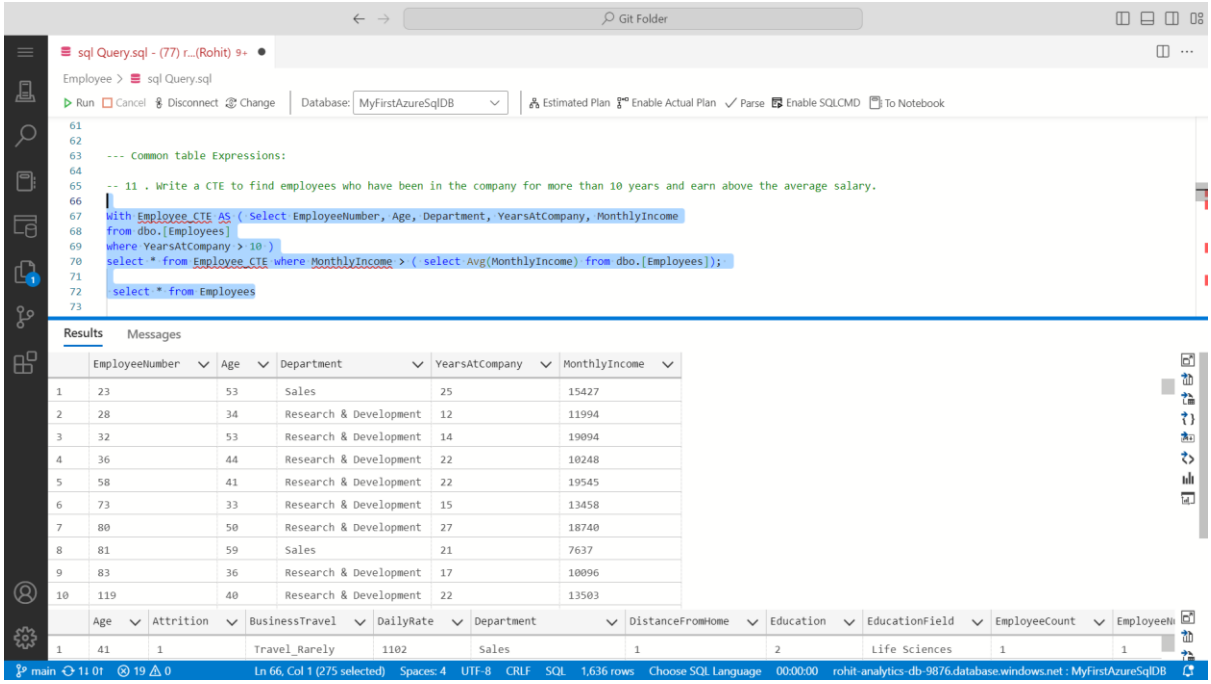
9. Minimum & Maximum Values: Find the youngest and oldest employee in each department.



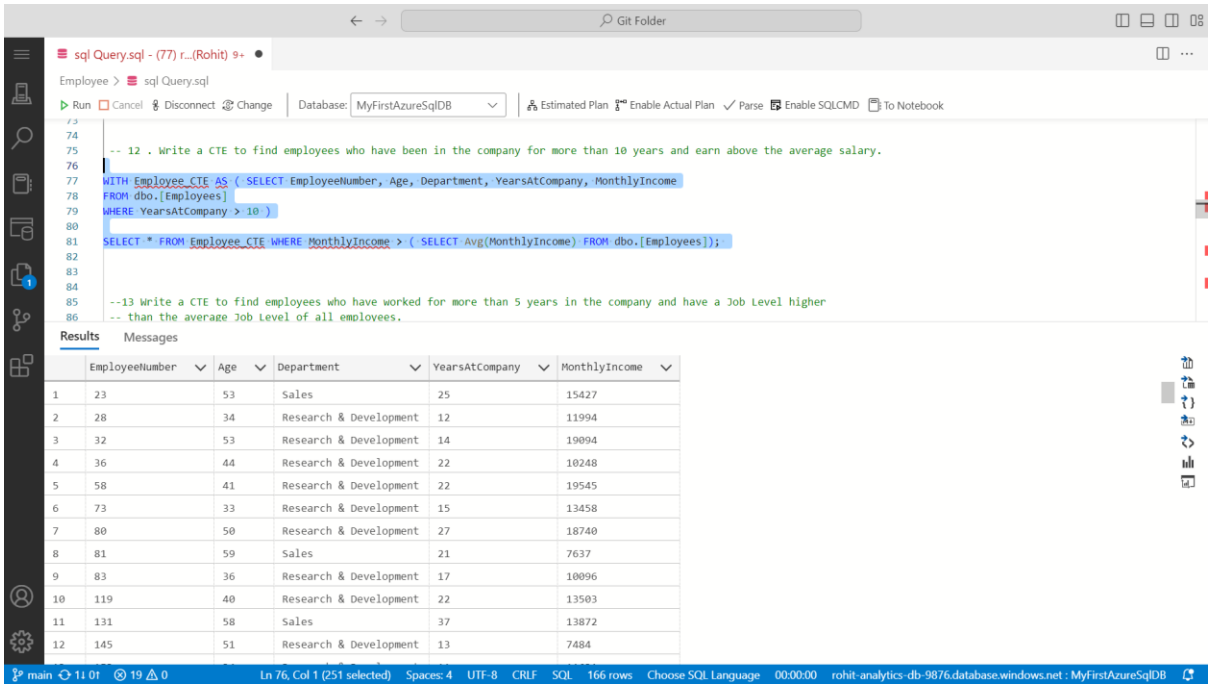
10. Checking Distinct Values: Count and list unique job roles in the dataset.



11. CTE - Salary & Experience: Find employees with over 10 years of experience earning above the average salary.



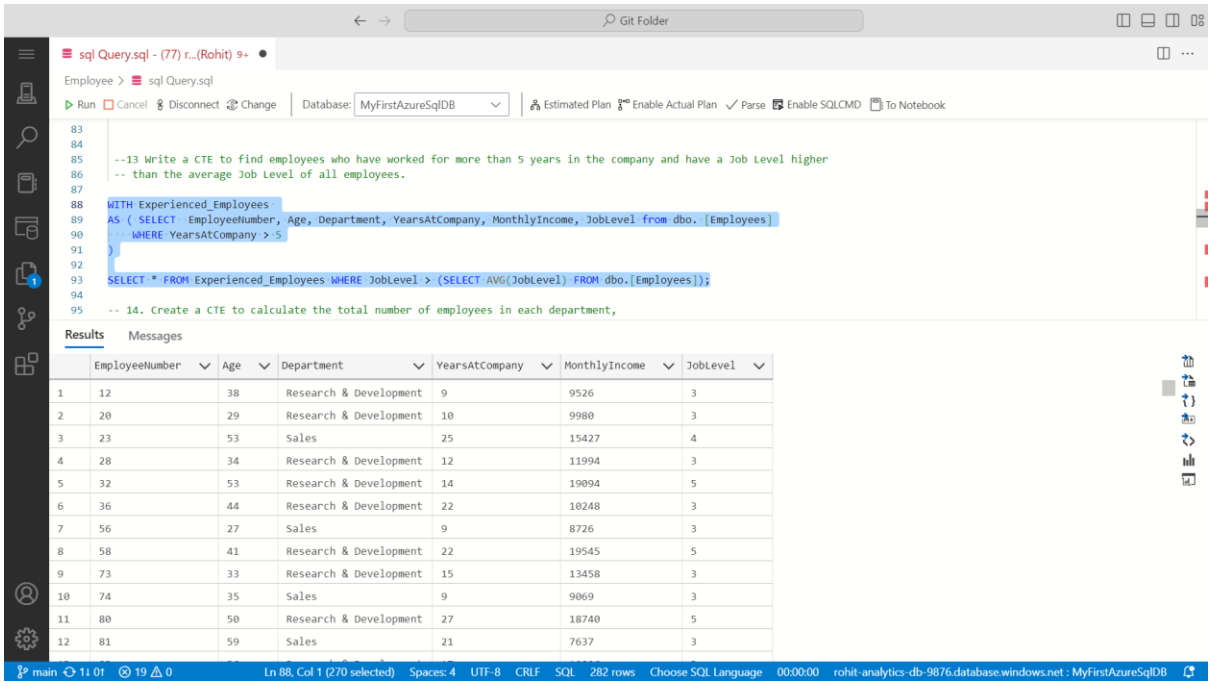
12. CTE - Experience & Salary: Repeat the above with a refined query.



```sql
-- 12 . Write a CTE to find employees who have been in the company for more than 10 years and earn above the average salary.

WITH Employee_CTE AS ( SELECT EmployeeNumber, Age, Department, YearsAtCompany, MonthlyIncome
FROM dbo.[Employees]
WHERE YearsAtCompany > 10 )

SELECT * FROM Employee_CTE WHERE MonthlyIncome > ( SELECT Avg(MonthlyIncome) FROM dbo.[Employees]);



--13 Write a CTE to find employees who have worked for more than 5 years in the company and have a Job Level higher
-- than the average Job Level of all employees.
```

	EmployeeNumber	Age	Department	YearsAtCompany	MonthlyIncome
1	23	53	Sales	25	15427
2	28	34	Research & Development	12	11994
3	32	53	Research & Development	14	19094
4	36	44	Research & Development	22	10248
5	58	41	Research & Development	22	19545
6	73	33	Research & Development	15	13458
7	80	50	Research & Development	27	18740
8	81	59	Sales	21	7637
9	83	36	Research & Development	17	10096
10	119	40	Research & Development	22	13503
11	131	58	Sales	37	13872
12	145	51	Research & Development	13	7484

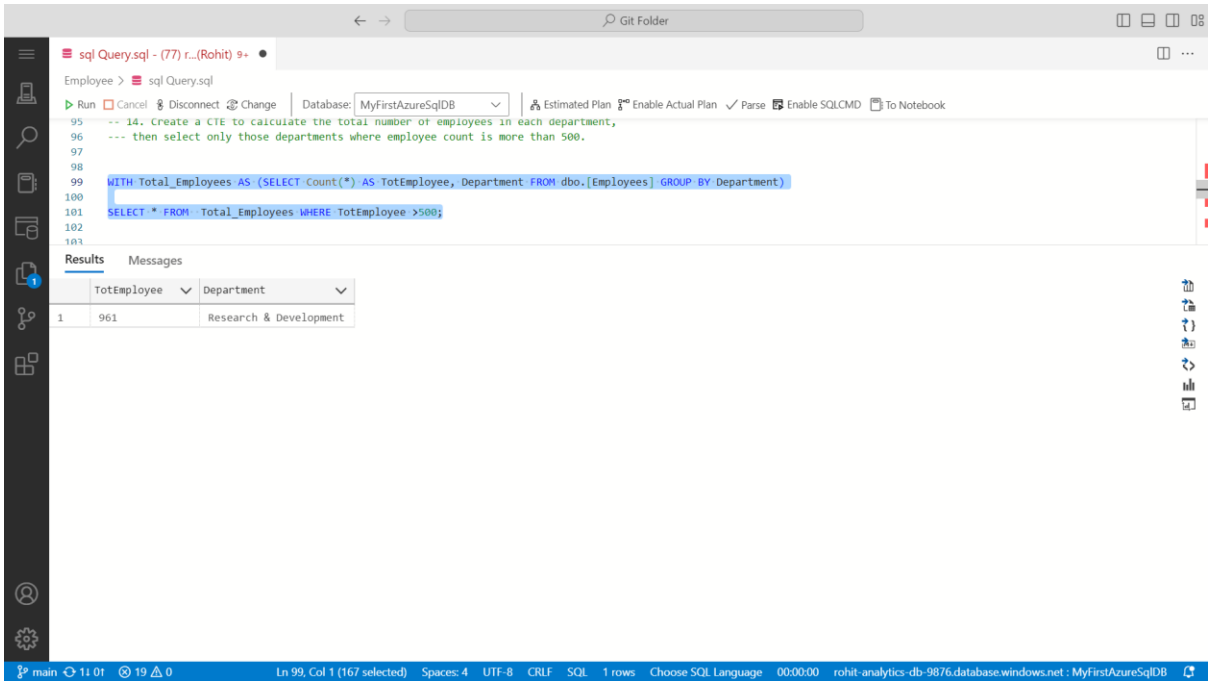
13. CTE - Job Level: Find employees with more than 5 years of experience and a Job Level higher than the average.



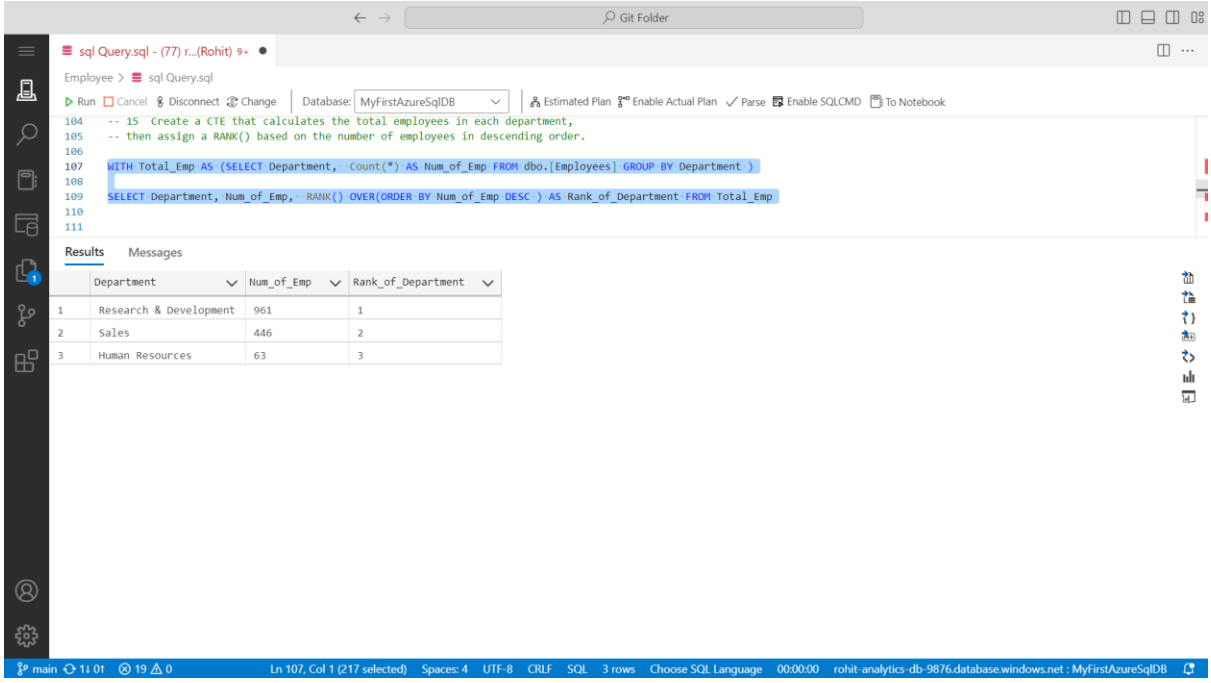
```sql
--13 Write a CTE to find employees who have worked for more than 5 years in the company and have a Job Level higher
-- than the average Job Level of all employees.

WITH Experienced_Employees
AS ( SELECT EmployeeNumber, Age, Department, YearsAtCompany, MonthlyIncome, JobLevel from dbo. [Employees]
    WHERE YearsAtCompany > 5
)

SELECT * FROM Experienced_Employees WHERE JobLevel > (SELECT AVG(JobLevel) FROM dbo.[Employees]);

-- 14. Create a CTE to calculate the total number of employees in each department,
```

	EmployeeNumber	Age	Department	YearsAtCompany	MonthlyIncome	JobLevel
1	12	38	Research & Development	9	9526	3
2	20	29	Research & Development	10	9980	3
3	23	53	Sales	25	15427	4
4	28	34	Research & Development	12	11994	3
5	32	53	Research & Development	14	19094	5
6	36	44	Research & Development	22	10248	3
7	56	27	Sales	9	8726	3
8	58	41	Research & Development	22	19545	5
9	73	33	Research & Development	15	13458	3
10	74	35	Sales	9	9069	3
11	80	50	Research & Development	27	18740	5
12	81	59	Sales	21	7637	3

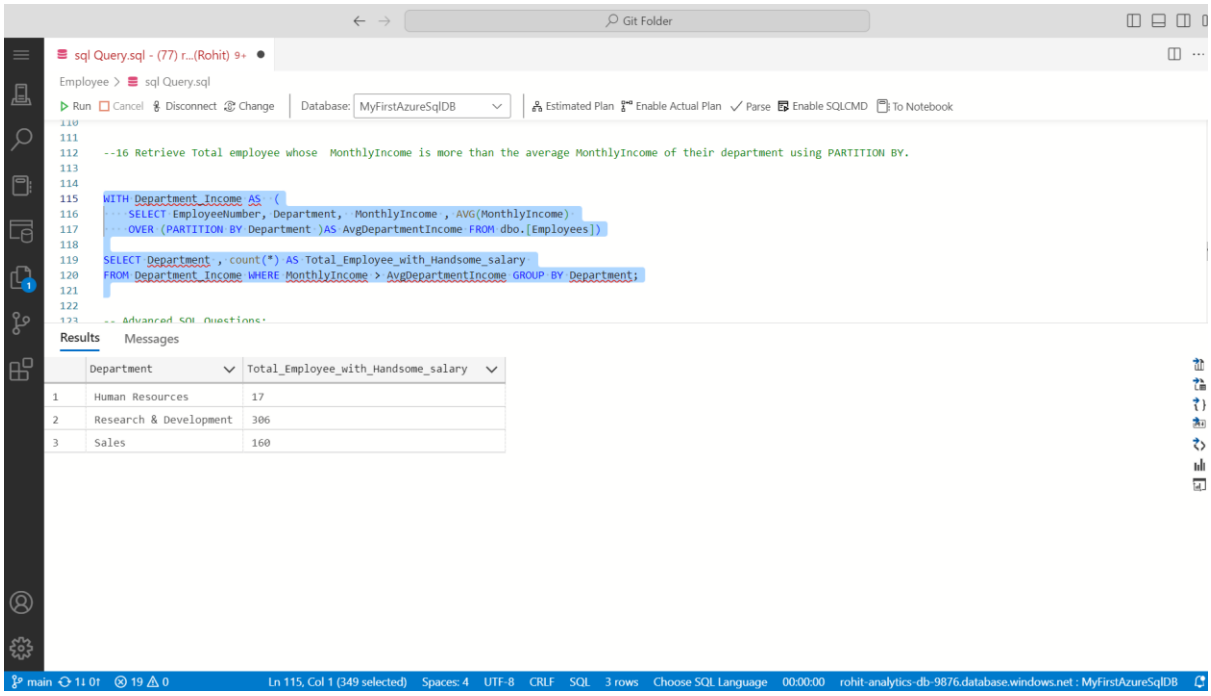
14. CTE - Employee Count: Find departments with more than 500 employees.



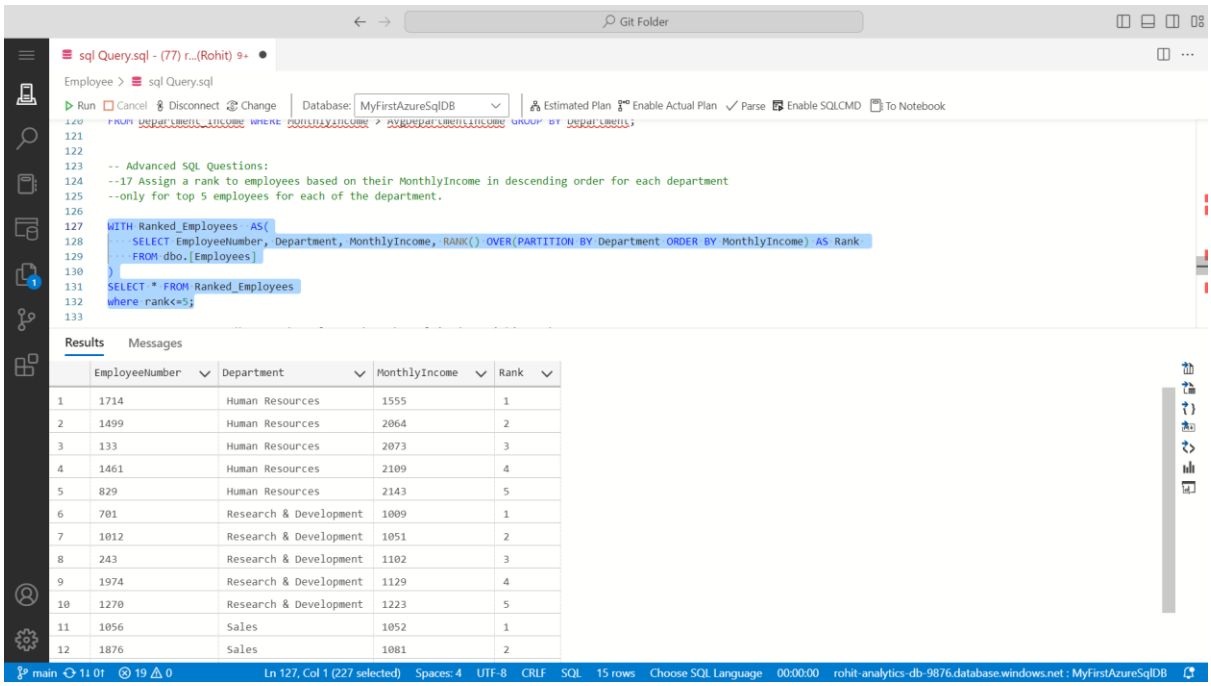
15. CTE - Ranking Departments: Rank departments by total employees in descending order.



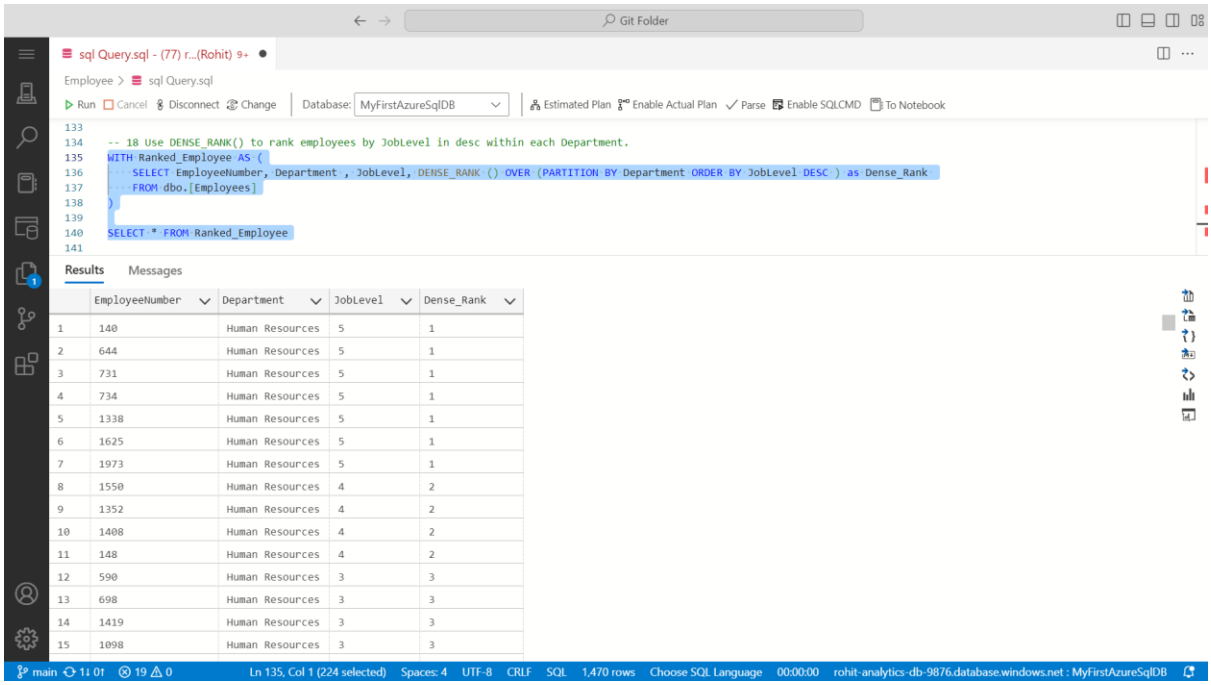
16. PARTITION BY - Income: Count employees earning above their department's average salary.



17. RANK() - Top Earners: Rank the top 5 highest-paid employees in each department.



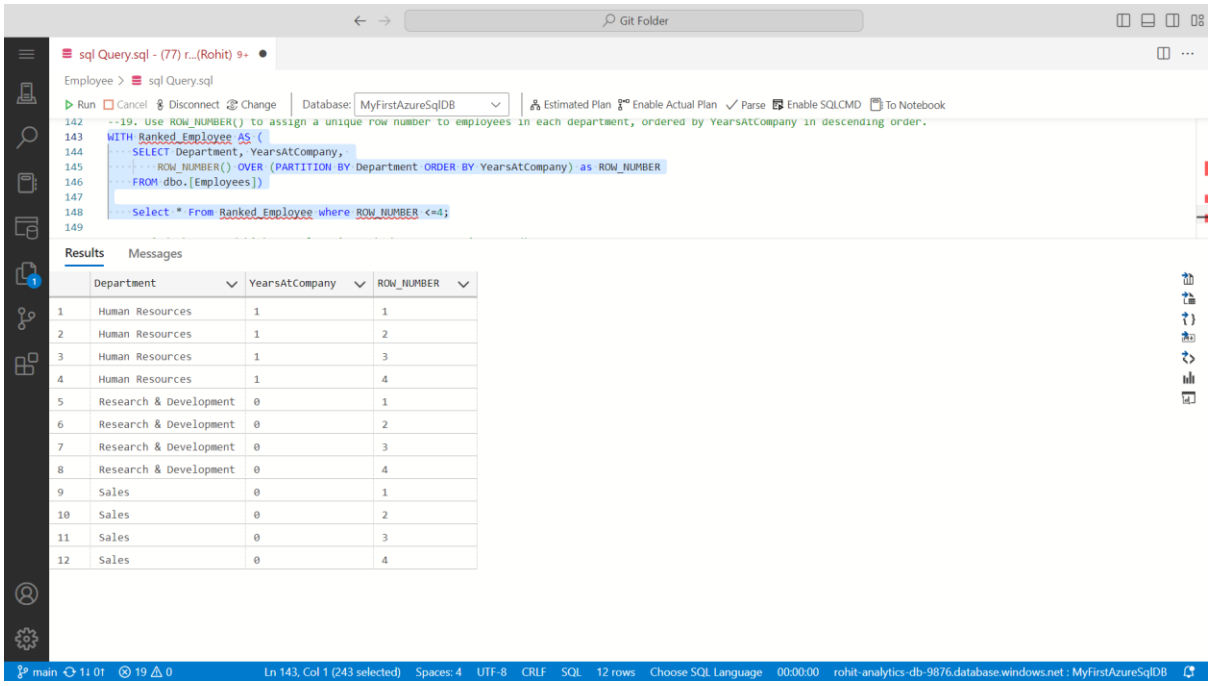
18. DENSE_RANK() - Job Level: Rank employees by Job Level within each department.



```sql
-- 18 Use DENSE_RANK() to rank employees by JobLevel in desc within each Department.
WITH Ranked_Employee AS (
    SELECT EmployeeNumber, Department , JobLevel, DENSE_RANK () OVER (PARTITION BY Department ORDER BY JobLevel DESC ) as Dense_Rank
    FROM dbo.[Employees]
)

SELECT * FROM Ranked_Employee
```

	EmployeeNumber	Department	JobLevel	Dense_Rank
1	140	Human Resources	5	1
2	644	Human Resources	5	1
3	731	Human Resources	5	1
4	734	Human Resources	5	1
5	1338	Human Resources	5	1
6	1625	Human Resources	5	1
7	1973	Human Resources	5	1
8	1550	Human Resources	4	2
9	1352	Human Resources	4	2
10	1408	Human Resources	4	2
11	148	Human Resources	4	2
12	590	Human Resources	3	3
13	698	Human Resources	3	3
14	1419	Human Resources	3	3
15	1098	Human Resources	3	3

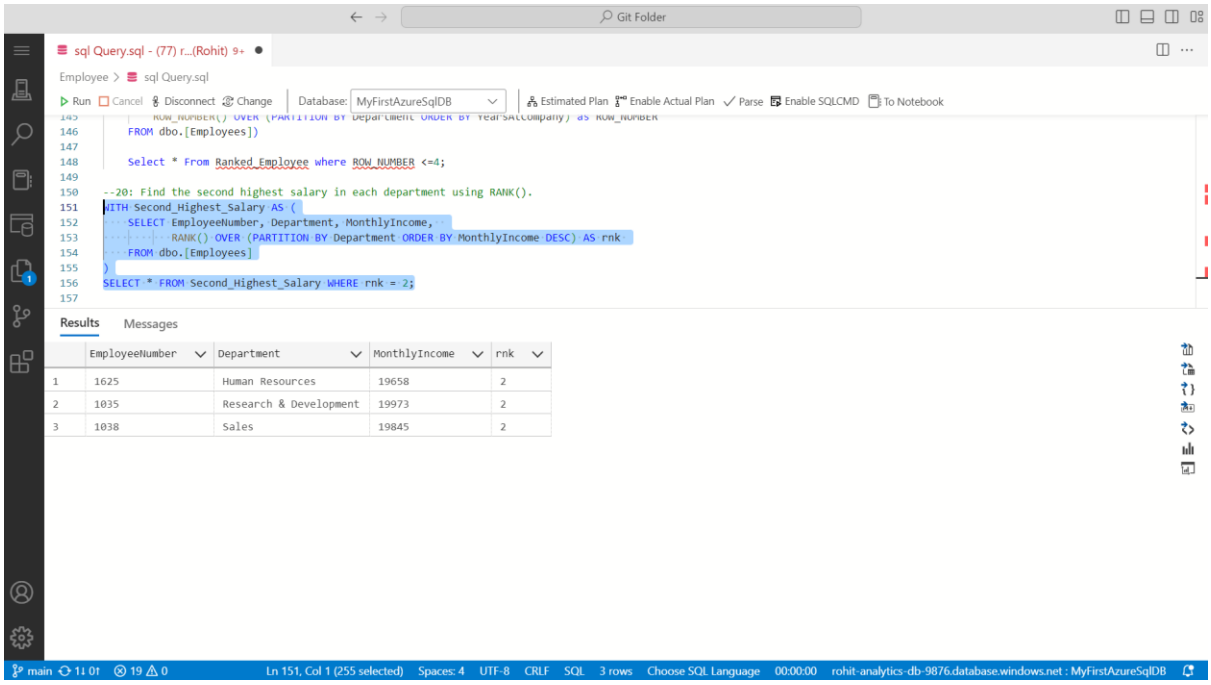
19. ROW_NUMBER() - Tenure: Assign row numbers to employees in each department based on YearsAtCompany.



```sql
--19. Use ROW_NUMBER() to assign a unique row number to employees in each department, ordered by YearsAtCompany in descending order.
WITH Ranked_Employee AS (
    SELECT Department, YearsAtCompany,
        ROW_NUMBER() OVER (PARTITION BY Department ORDER BY YearsAtCompany) as ROW_NUMBER
    FROM dbo.[Employees])

    Select * From Ranked_Employee where ROW_NUMBER <=4;
```

	Department	YearsAtCompany	ROW_NUMBER
1	Human Resources	1	1
2	Human Resources	1	2
3	Human Resources	1	3
4	Human Resources	1	4
5	Research & Development	0	1
6	Research & Development	0	2
7	Research & Development	0	3
8	Research & Development	0	4
9	Sales	0	1
10	Sales	0	2
11	Sales	0	3
12	Sales	0	4

20. RANK() - Second Highest Salary: Find the second-highest salary in each department.



21. RANK()/DENSE_RANK() - Top 3 Earners: Find the top 3 highest-paid employees in each department.