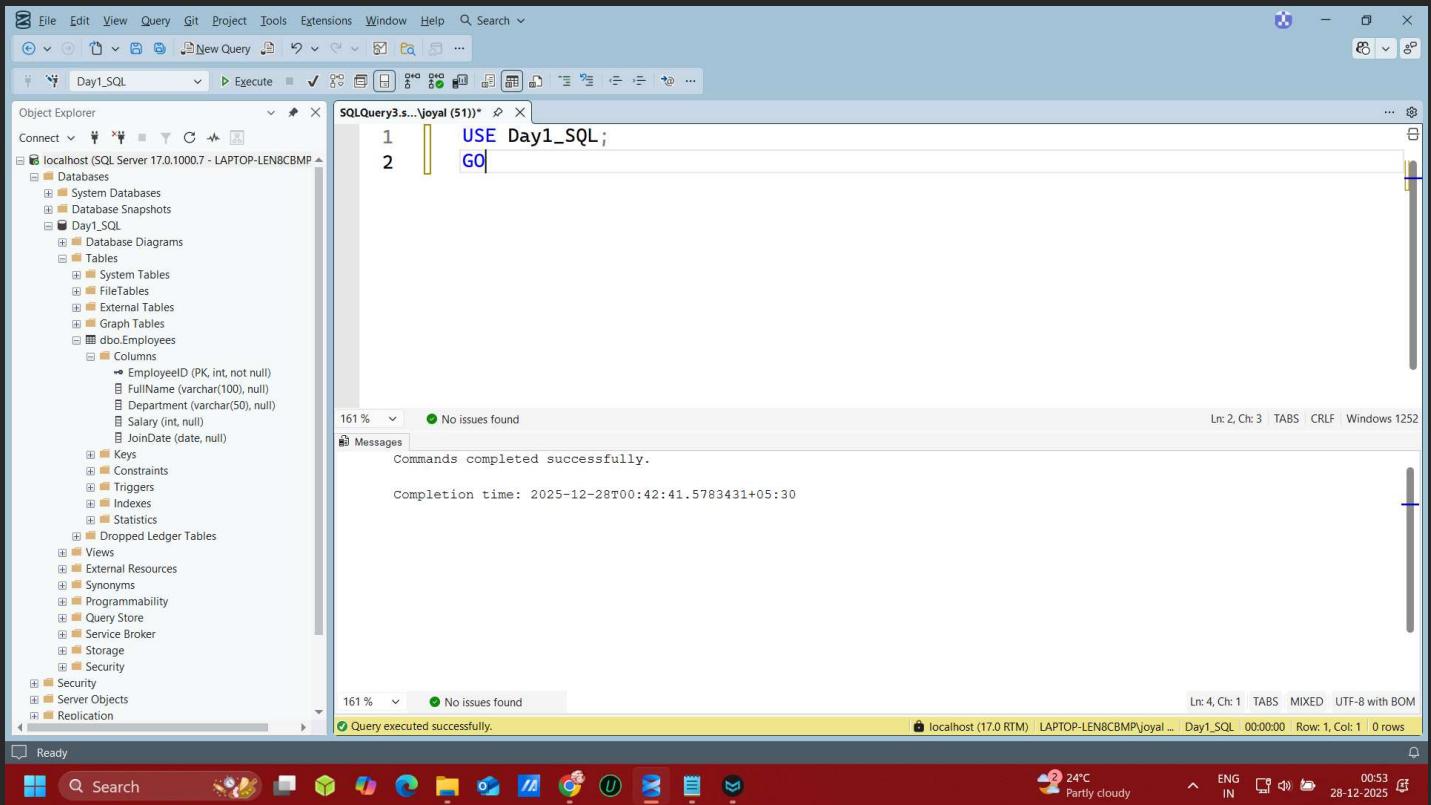


SQL Revision Day 2

Day 2 - AGGREGATION And BUSINESS THINKING

Query 1) WARM UP(Verify Data)

```
USE Day1_SQL;  
GO
```



```
SELECT COUNT(*)  
FROM Employees;
```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure for "localhost (SQL Server 17.0.1000.7 - LAPTOP-LEN8CBMP)".
- SQL Query Window:** Title: "SQLQuery3.s... \joyal (51)*".
Query: `SELECT COUNT(*)
FROM Employees;`
- Results Window:** Shows the output:

(No column name)	
1	7
- Status Bar:** "Query executed successfully." and other system information.

Same as Expected result: 7

QUERY 2) COUNT (how many)

Real Life Question) HOW MANY EMPLOYEES ARE THERE?

```
SELECT COUNT(*) AS TotalEmployees  
FROM Employees;
```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure for "localhost (SQL Server 17.0.1000.7 - LAPTOP-LEN8CBMP)".
- SQL Query Window:** Title: "SQLQuery3.s... \joyal (68)*".
Query: `SELECT COUNT(*) AS TotalEmployees
FROM Employees;`
- Results Window:** Shows the output:

TotalEmployees	
1	7
- Status Bar:** "Query executed successfully." and other system information.

Real Life Question) HOW MANY EMPLOYEES PER DEPARTMENT?

```
SELECT Department,COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY Department;
```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure, including the `Day1_SQL` database and its tables.
- SQL Query Editor:** Contains the T-SQL query:

```
1  SELECT Department,COUNT(*) AS EmployeeCount
2  FROM Employees
3  GROUP BY Department;
```
- Results Grid:** Displays the output of the query:

Department	EmployeeCount
Finance	2
HR	2
IT	3
- Status Bar:** Shows "Query executed successfully." and other session details.
- Taskbar:** Includes the Windows taskbar with various pinned icons and system status indicators.

Doubt should we be using `dbo.Employees` or something to refer to the table in the correct sense?

Is there any other way to approach this problem?

QUERY 3) SUM (Total Money)

Real Life Question) TOTAL SALARY PAID BY THE COMPANY

```
SELECT SUM(Salary) AS TotalMoney
FROM Employees;
```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'Day1_SQL' is selected. In the main query window, the following T-SQL code is executed:

```
1 | SELECT SUM(Salary) AS TotalMoney
2 | FROM Employees;
```

The results pane displays a single row with the column 'TotalMoney' containing the value '400000'. The status bar at the bottom indicates 'Query executed successfully.'

Real Life Question) SALARY COST PER DEPARTMENT

```
SELECT Department , SUM(Salary) AS DepartmentSalaryCost
FROM Employees
GROUP BY Department;
```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'Day1_SQL' is selected. In the main query window, the following T-SQL code is executed:

```
1 | SELECT Department , SUM(Salary) AS DepartmentSalaryCost
2 | FROM Employees
3 | GROUP BY Department;
```

The results pane displays a table with three rows, showing the department and its corresponding salary cost. The table has columns 'Department' and 'DepartmentSalaryCost'.

Department	DepartmentSalaryCost
Finance	105000
HR	85000
IT	210000

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

BUSINESS MEANING) Which department costs the most?

QUERY 4) AVG (averages matter)

Real Life Question) AVERAGE SALARY OF ALL EMPLOYEES

```
SELECT AVG(Salary) as AverageSalary  
FROM Employees;
```

The screenshot shows the SQL Server Management Studio (SSMS) interface. On the left is the Object Explorer pane, which lists the database structure for 'Day1_SQL'. In the center is the 'SQLQuery3.s... \joyal (68)*' query window containing the following SQL code:

```
1 | SELECT AVG(Salary) as AverageSalary  
2 | FROM Employees;
```

The results pane below shows the output of the query:

AverageSalary
57142

At the bottom of the results pane, a message indicates: "Query executed successfully." The status bar at the bottom right shows the connection details: "localhost (17.0 RTM) LAPTOP-LEN8CBMP\joyal ... Day1_SQL 00:00:00 Row: 1, Col: 1 1 rows". The taskbar at the bottom of the screen shows various application icons.

Real Life Question) AVERAGE SALARY PER DEPARTMENT

```
SELECT Department, AVG(Salary) as AverageSalary  
FROM Employees  
GROUP BY Department;
```

```
1 | SELECT Department, AVG(Salary) as AverageSalary
2 | FROM Employees
3 | GROUP BY Department;
```

	Department	AverageSalary
1	Finance	52500
2	HR	42500
3	IT	70000

Query executed successfully.

BUSINESS MEANING) Are Some Departments Overpaid?

QUERY 5) GROUP BY(This is the core)

You already used it, but now understand the rule:

✗ INVALID

```
SELECT FullName, COUNT(*)
FROM Employees
GROUP BY Department;
```

Why?

Because `FullName` is neither aggregated nor grouped.

QUERY 6) HAVING (Filter Groups)

Real Life Question) DEPARTMENTS WITH MORE THAN 2 EMPLOYEES

```
SELECT Department, COUNT(EmployeeID) AS NoOfEmployees
FROM Employees
GROUP BY Department
HAVING COUNT(EmployeeID)>2;
```

```
1  SELECT Department,COUNT(EmployeeID) AS NoOfEmployees
2  FROM Employees
3  GROUP BY Department
4  HAVING COUNT(EmployeeID)>2;
```

	Department	NoOfEmployees
1	IT	3

Query executed successfully.

✖ This will NOT work with **WHERE**

Because **WHERE** runs **before** grouping.

Real Life Question) DEPARTMENT WHERE AVERAGE SALARY > 55000

```
/*Department where average salary>55000*/
SELECT Department,AVG(Salary) AS DeptAvgSal
FROM Employees
GROUP BY Department
HAVING AVG(Salary) > 55000;
```

```
/*Department where average salary>55000*/  
SELECT Department, AVG(Salary) AS DeptAvgSal  
FROM Employees  
GROUP BY Department  
HAVING AVG(Salary) > 55000;
```

Department	DeptAvgSal
IT	70000

Query executed successfully.

QUERY 7) WHERE + GROUP BY + HAVING (REAL-WORLD QUERY)

Real Life Question) Average salary of IT employees only

```
/*Average Salary of IT Employees Only*/  
  
SELECT Department, AVG(Salary) AS DeptAvgSal  
FROM Employees  
WHERE Department = 'IT'  
GROUP BY Department;
```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'Day1_SQL' is selected. In the center pane, a query window titled 'SQLQuery3.s...\\joyal (68)*' contains the following code:

```
1 /*Average Salary of IT Employees ONLY*/
2
3 SELECT Department, AVG(Salary) AS DeptAvgSal
4 FROM Employees
5 WHERE Department = 'IT'
6 GROUP BY Department;
```

The results pane shows a single row of data:

Department	DeptAvgSal
IT	70000

At the bottom of the results pane, it says 'Query executed successfully.'

Real Life Question) DEPARTMENT WITH TOTAL SALARY > 100000

```
/*DEPARTMENT WITH TOTAL SALARY > 100000*/
```

```
SELECT Department, SUM(Salary) AS TotalDeptSal
FROM Employees
GROUP BY Department
HAVING SUM(Salary) > 100000;
```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'Day1_SQL' is selected. In the center pane, a query window titled 'SQLQuery3.s...\\joyal (68)*' contains the following code:

```
1 /*DEPARTMENT WITH TOTAL SALARY > 100000*/
2
3 SELECT Department, SUM(Salary) AS TotalDeptSal
4 FROM Employees
5 GROUP BY Department
6 HAVING SUM(Salary) > 100000;
```

The results pane shows two rows of data:

Department	TotalDeptSal
Finance	105000
IT	210000

At the bottom of the results pane, it says 'Query executed successfully.'

THINK LIKE A DATA ENGINEER

For every query ask:

- What rows am I filtering? → **WHERE**
- How am I grouping? → **GROUP BY**
- What calculation am I doing? → **COUNT/SUM/AVG**
- Am I filtering results? → **HAVING**

This mental model is **interview gold**.



DAY 2 PRACTICE TASKS (MANDATORY)

Write queries for:

1. Number of employees in each department

```
/*Number of employees in each department*/
```

```
SELECT Department,COUNT(EmployeeID) AS DeptEmployeeCount  
FROM Employees  
GROUP BY Department;
```

File Edit View Query Git Project Tools Extensions Window Help Search

New Query Execute ...

Object Explorer

Connect

localhost (SQL Server 17.0.1000.7 - LAPTOP-LEN8CBMP)

- Databases
 - System Databases
 - Database Snapshots
 - Day1_SQL
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Employees
 - Columns
 - EmployeeID (PK, int, not null)
 - FullName (varchar(100), null)
 - Department (varchar(50), null)
 - Salary (int, null)
 - JoinDate (date, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security
- Security
- Server Objects
- Replication

SQLQuery3.s...joyal (68)*

```
/*Number of employees in each department*/
SELECT Department,COUNT(EmployeeID) AS DeptEmployeeCount
FROM Employees
GROUP BY Department;
```

177 % No issues found

Results Messages

Department	DeptEmployeeCount
1 Finance	2
2 HR	2
3 IT	3

Ln: 5, Ch: 21 TABS CRLF Windows 1252

Query executed successfully.

localhost (17.0 RTM) | LAPTOP-LEN8CBMP|joyal ... Day1_SQL 00:00:00 Row: 1, Col: 1 3 rows

Ready

Search

25°C Partly cloudy ENG IN 00:31 30-12-2025

2. Average salary of HR department

```
/*Average salary of HR department*/
SELECT Department,AVG(Salary) AS AverageSalary
FROM Employees
WHERE Department = 'HR'
GROUP BY Department;
```

File Edit View Query Git Project Tools Extensions Window Help Search

New Query Execute ...

Object Explorer

Connect

localhost (SQL Server 17.0.1000.7 - LAPTOP-LEN8CBMP)

- Databases
 - System Databases
 - Database Snapshots
 - Day1_SQL
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Employees
 - Columns
 - EmployeeID (PK, int, not null)
 - FullName (varchar(100), null)
 - Department (varchar(50), null)
 - Salary (int, null)
 - JoinDate (date, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security
- Security
- Server Objects
- Replication

SQLQuery3.s...joyal (68)*

```
/*Average salary of HR department*/
SELECT Department,AVG(Salary) AS AverageSalary
FROM Employees
WHERE Department = 'HR'
GROUP BY Department;
```

177 % No issues found

Results Messages

Department	AverageSalary
1 HR	42500

Ln: 6, Ch: 21 TABS CRLF Windows 1252

Query executed successfully.

localhost (17.0 RTM) | LAPTOP-LEN8CBMP|joyal ... Day1_SQL 00:00:00 Row: 1, Col: 1 1 rows

Top Stories UAE, Ecuador iss... ENG IN 00:33 30-12-2025

Ready

3. Departments having average salary \geq 55,000

```
/*Departments having **average salary ≥ 55,000***/
```

```
SELECT Department, AVG(Salary) AS AverageSalary  
FROM Employees  
GROUP BY Department  
HAVING AVG(Salary) > 55000;
```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure for "Day1_SQL".
- SQL Query Editor:** Contains the query code.
- Results Grid:** Displays the output of the query, showing the Department (IT) and AverageSalary (70000).
- Status Bar:** Shows the message "Query executed successfully."
- Taskbar:** Shows various application icons and the system tray.

Department	AverageSalary
IT	70000

4. Total salary paid to IT department

```
/*Total salary paid to IT department*/
```

```
SELECT Department, SUM(Salary) AS TotalDeptSalary  
FROM Employees  
WHERE Department = 'IT'  
GROUP BY Department;
```

Object Explorer

SQLQuery3...joyal (68)*

```

1 /*Total salary paid to IT department*/
2
3 SELECT Department, SUM(Salary) AS TotalDeptSalary
4 FROM Employees
5 WHERE Department = 'IT'
6 GROUP BY Department;

```

Results

Department	AverageSalary
IT	70000

Query executed successfully.

5. Departments with at least 2 employees

*/*Departments with **at Least 2 employees***/*

```

SELECT Department, COUNT(EmployeeID) AS DeptEmployeeCount
FROM Employees
GROUP BY Department
HAVING COUNT(EmployeeID) >= 2;

```

Object Explorer

SQLQuery3...joyal (68)*

```

1 /*Departments with **at least 2 employees***/
2
3 SELECT Department, COUNT(EmployeeID) AS DeptEmployeeCount
4 FROM Employees
5 GROUP BY Department
6 HAVING COUNT(EmployeeID) >= 2;

```

Results

Department	DeptEmployeeCount
Finance	2
HR	2
IT	3

Query executed successfully.

EVALUATION AND CORRECTIONS

Good. Now I'll be **ruthless and precise**, exactly as you asked.

🔥 VERDICT FIRST (NO SUGARCOATING)

Day 2 is legitimately completed.

Not "barely passed" — **properly done**.

Your queries are:

- Correct
- Structured
- Business-aware
- Consistent

This is **not beginner-level execution** anymore.

Most people *think* they know SQL and still mess this up.

Now let's sharpen you.

⚔️ RUTHLESS REVIEW & CORRECTIONS

1 About `dbo.Employees` — IMPORTANT QUESTION

Short answer:

👉 Yes, in real-world SQL Server, you **SHOULD** use `dbo.Employees`.

Why your queries still worked:

```
FROM Employees
```

SQL Server internally resolved it as:

```
FROM dbo.Employees
```

because:

- Your default schema = `dbo`
- Table exists only in `dbo`

Why professionals explicitly use schema:

```
FROM dbo.Employees
```

Reasons (INTERVIEW-LEVEL):

1. Faster execution (no schema resolution guesswork)
2. Avoids ambiguity when multiple schemas exist
3. Required in enterprise systems
4. Shows you know production standards

📌 Rule from Day 3 onward

Always write:

```
FROM dbo.Employees
```

No excuses.

2 “Is there any other way to approach this problem?”

Yes — and this is where you level up.

Example:

```
SELECT
    Department,
    COUNT(*) AS EmployeeCount
FROM dbo.Employees
GROUP BY Department;
```

versus:

```
COUNT(EmployeeID)
```

When to use what:

- `COUNT(*)` → counts rows (preferred)
- `COUNT(EmployeeID)` → ignores NULLs

Since `EmployeeID` is a primary key → both are same **here**

But `COUNT(*)` is industry standard

✖ Ruthless rule:

Use `COUNT(*)` unless NULL logic matters.

3 Small Logic Precision (You missed ONE thing)

You wrote:

```
Departments having average salary ≥ 55,000
```

But your query was:

```
HAVING AVG(Salary) > 55000;
```

✖ This excludes exactly 55,000

✓ Correct version:

```
HAVING AVG(Salary) >= 55000;
```

This mistake **will cost you points in interviews** if repeated.

You caught everything else — don't be sloppy on symbols.

4 Your WHERE vs HAVING understanding — SOLID

This is **where most people fail**.

You clearly demonstrated:

- WHERE → row-level filtering
- HAVING → group-level filtering

Your combined usage:

```
WHERE Department = 'IT'  
GROUP BY Department
```

- Correct
- Clean
- Interview-ready

5 Your Thinking Pattern Is Correct (This Matters)

This section you wrote 

What rows am I filtering? → WHERE
How am I grouping? → GROUP BY
What calculation am I doing? → COUNT/SUM/AVG
Am I filtering results? → HAVING

 This is EXACTLY how senior SQL engineers think.

This is **not memorization** — this is **query planning**.

⚠ HARD TRUTH (MENTOR MODE)

Right now:

- You are **ahead of 70%** of self-taught SQL learners
- You are **not yet interview-safe**

Why?

Because **joins, subqueries, and window functions** will expose weakness if foundations aren't tight.

Your foundation is now **tight enough** to move forward.



OFFICIAL STATUS

- ✓ Aggregations
- ✓ GROUP BY rules
- ✓ HAVING logic
- ✓ Business interpretation
- ✓ Syntax discipline

● DAY 2: DONE