🎯 Aggregation Functions – Complete Concept (for Learning & Interviews)

1️⃣ Aggregation Functions అంటే ఏమిటి?

👉 Aggregation functions అనేవి multiple rows పై calculations చేసి ఒకే single value return చేసే functions.

ఉదా:

మొత్తం జీతం (Total salary)

సగటు జీతం (Average salary)

ఎకుఫ మారుఫు (Highest marks)

తకుఫ మారుఫు (Lowest marks)

మొత్తం స్టూడెంట్స్సంఖ్య(Total count)

🔷 Common Aggregation Functions

Function	Use (ఉపయోగం)	Example (ఉదాహరణ)
SUM()	మొత్తం విలువ	SUM(salary)
AVG()	సగటు విలువ	AVG(salary)
COUNT()	రికార్డుల సంఖ్య	COUNT(*)
MAX()	గరిష్ట విలువ	MAX(marks)
MIN()	కనిష్ట విలువ	MIN(marks)

📦 2️⃣ Table Example (Student Table)

```
CREATE TABLE student (
    student_id SERIAL PRIMARY KEY,
    student_name VARCHAR(50),
    dept_name VARCHAR(50),
    marks INT,
    fees NUMERIC(10,2)
);

INSERT INTO student (student_name, dept_name, marks, fees) VALUES
('Ravi', 'CSE', 85, 35000),
('Sneha', 'CSE', 90, 36000),
('Arjun', 'ECE', 78, 34000),
('Deepa', 'ECE', 88, 33000),
('Kiran', 'EEE', 60, 32000),
('Lalitha', 'EEE', 72, 31000),
('Manoj', 'MECH', 55, 30000),
('Sujatha', 'MECH', 65, 31000);
```

💡 3️⃣ Basic Aggregation Queries

✅ a) Total Marks

```
SELECT SUM(marks) AS total_marks FROM student;
```

Result: → మొత్తం మార్కులు: 593

✅ b) Average Marks

```
SELECT AVG(marks) AS avg_marks FROM student;
```

Result: → సగటు మార్కులు: 74.12

✅ c) Number of Students

SELECT COUNT(*) AS total_students FROM student;

Result: → మొత్తం 8 మంది

✅ d) Highest Marks

SELECT MAX(marks) AS highest_marks FROM student;

Result: → 90

✅ e) Lowest Marks

SELECT MIN(marks) AS lowest_marks FROM student;

Result: → 55

🧠 4️⃣ Real-time Scenarios

Requirement	**Query**	**Description**
ప్రతి department లో average marks	SELECT dept_name, AVG(marks) FROM student GROUP BY dept_name;	Department-wise performance
Highest fee paid by student	SELECT MAX(fees) FROM student;	Finance report
Total fees collected	SELECT SUM(fees) FROM student;	Accounts summary
Students count in each dept	SELECT dept_name, COUNT(*) FROM student GROUP BY dept_name;	Dept strength report
Lowest marks student info	SELECT * FROM student WHERE marks = (SELECT MIN(marks) FROM student);	Identify weak student

⌨ 5️⃣ Visualization Diagram

Student Marks

Ravi | 85

Sneha | 90

Arjun | 78

Deepa | 88

Kiran | 60

Lalitha | 72

Manoj | 55

Sujatha | 65

SUM() → 593

AVG() → 74.12

MAX() → 90

MIN() → 55

COUNT() → 8

6️⃣ Real-time College / Company Reports

Use Case	**Example Function**	**Real-time Meaning**
Students total strength	COUNT()	College admission report
Average marks per dept	AVG()	Academic performance
Total fees collection	SUM()	Finance department report
Highest marks per subject	MAX()	Topper list
Lowest marks	MIN()	Identify weak students

7️⃣ Interview-Level Questions

🔰 Beginner Level:

1. What are aggregation functions in SQL?

👉 Functions that perform a calculation on a set of values and return a single value.

2. Name a few aggregation functions.

👉 SUM(), AVG(), COUNT(), MAX(), MIN().

3. Can we use WHERE with aggregation functions?

❌ No, WHERE works before aggregation; use HAVING for groups.

🔷 Intermediate Level:

1. Difference between COUNT(column) and COUNT(*)?

👉 COUNT(column) skips NULL values,

👉 COUNT(*) counts all rows including NULLs.

2. Can we use aggregation functions without GROUP BY?

✅ Yes, on entire table (total summary).

3. How to find department-wise maximum marks?

```
SELECT dept_name, MAX(marks)
FROM student
GROUP BY dept_name;
```

🔷 Expert Level:

1. How to get the 2nd highest marks using aggregation?

```
SELECT MAX(marks)
FROM student
```

WHERE marks < (SELECT MAX(marks) FROM student);

2. How to count distinct departments?

SELECT COUNT(DISTINCT dept_name) FROM student;

3. Can we use multiple aggregation functions in one query?

✅ Example:

```
SELECT
    COUNT(*) AS total,
    AVG(marks) AS average,
    MAX(marks) AS top,
    MIN(marks) AS low
FROM student;
```

🧠 8️⃣ Concept Summary (Teaching Tip)

Function	Meaning	Real-time Use
SUM()	Total	Total fees, total salary
AVG()	Average	Average marks/salary
COUNT()	Count	Students/employees count
MAX()	Highest	Topper / Highest salary
MIN()	Lowest	Weak student / Lowest salary

📄 File: aggregation_functions_demo.sql (**PRACTICE PURPOSE**)

==========

```
-- FILE: aggregation_functions_demo.sql
-- TOPIC: SQL Aggregation Functions (SUM, AVG, COUNT, MAX, MIN)
-- FOR: Students / Beginners / Interview Preparation
--
================================================
1️⃣ DROP OLD TABLES IF EXIST
DROP TABLE IF EXISTS student;

2️⃣ CREATE TABLE
CREATE TABLE student (
    student_id SERIAL PRIMARY KEY,
    student_name VARCHAR(50),
    dept_name VARCHAR(50),
    marks INT,
    fees NUMERIC(10,2)
);
3️⃣ INSERT SAMPLE DATA
INSERT INTO student (student_name, dept_name, marks, fees) VALUES
('Ravi', 'CSE', 85, 35000),
('Sneha', 'CSE', 90, 36000),
('Arjun', 'ECE', 78, 34000),
('Deepa', 'ECE', 88, 33000),
('Kiran', 'EEE', 60, 32000),
('Lalitha', 'EEE', 72, 31000),
('Manoj', 'MECH', 55, 30000),
('Sujatha', 'MECH', 65, 31000);
 ================================================
4️⃣ BASIC AGGREGATION FUNCTIONS
================================================
```

a) Total Marks of all Students

```
SELECT SUM(marks) AS total_marks FROM student;
```

b) Average Marks

```
SELECT AVG(marks) AS avg_marks FROM student;
```

c) Number of Students

```
SELECT COUNT(*) AS total_students FROM student;
```

d) Highest Marks

```
SELECT MAX(marks) AS highest_marks FROM student;
```

e) Lowest Marks

```
SELECT MIN(marks) AS lowest_marks FROM student;
```

==========

5. MULTIPLE FUNCTIONS TOGETHER

==========

```
SELECT
    COUNT(*) AS total_students,
    SUM(marks) AS total_marks,
    AVG(marks) AS average_marks,
    MAX(marks) AS topper_marks,
    MIN(marks) AS lowest_marks
FROM student;
```

---

6. GROUP BY EXAMPLES (Real-time)

==========

Dept-wise average marks

```
SELECT dept_name, AVG(marks) AS avg_marks
FROM student
GROUP BY dept_name;
```

Dept-wise total students

```
SELECT dept_name, COUNT(*) AS total_students
FROM student
GROUP BY dept_name;
```

Dept-wise highest marks

```
SELECT dept_name, MAX(marks) AS topper_marks
FROM student
GROUP BY dept_name;
```

Dept-wise total fees collected

```
SELECT dept_name, SUM(fees) AS total_fees
FROM student
GROUP BY dept_name;
```

==========

7□HAVING CLAUSE EXAMPLES

==========

Departments with average marks greater than 75

```
SELECT dept_name, AVG(marks) AS avg_marks
FROM student
GROUP BY dept_name
HAVING AVG(marks) > 75;
```

Departments having more than 1 student

```
SELECT dept_name, COUNT(*) AS total_students
```

```
FROM student
GROUP BY dept_name
HAVING COUNT(*) > 1;
```

==========

8□INTERVIEW QUESTIONS & ANSWERS

==========

Q1. Difference between COUNT(*) and COUNT(column)?

COUNT(*) → Counts all rows (including NULL)

COUNT(column) → Skips NULL values in that column

```
SELECT COUNT(*), COUNT(marks) FROM student;
```

Q2. Find 2nd highest marks

```
SELECT MAX(marks) AS second_highest
FROM student
WHERE marks < (SELECT MAX(marks) FROM student);
```

Q3. Count distinct departments

```
SELECT COUNT(DISTINCT dept_name) AS total_departments FROM
student;
```

Q4. Combine multiple aggregations

```
SELECT
    AVG(marks) AS avg_marks,
    SUM(fees) AS total_fees,
    MAX(marks) AS max_marks,
    MIN(marks) AS min_marks
FROM student;
```

Q5. Real-time College Reports

a) Average marks per department

b) Total fees collected by each department

c) Departments with more than 2 students

All above combine GROUP BY + HAVING

==========

9️⃣ INTERVIEW TIPS (as Comments)

==========

🔷 Aggregation Functions always return a single value.

🔷 GROUP BY is used to apply aggregations on each group.

🔷 HAVING is used to filter the grouped results.

🔷 WHERE filters rows BEFORE grouping.

🔷 HAVING filters AFTER grouping.

🔷 You can use multiple aggregation functions in one query.

🔷 Always use GROUP BY before HAVING and ORDER BY at the end.

==========

✅ END OF FILE