

## SQL Aggregate Functions

টেবিল ডেটা:

**teacher2 Table:**

T\_id T\_name T\_dept T\_salary

201	Azad	CSE	1000
202	Kamal	EEE	2000
203	Hasan	CE	1000
204	Kamal	EEE	1000
205	Rony	CSE	2000
206	Rakib	CSE	4000

---

**Aggregate Functions কী?**

Aggregate Functions হলো এমন functions যা একাধিক rows-এর উপর calculation করে এবং একটি মাত্র result return করে।

**মূল Aggregate Functions:**

1. **COUNT()** - গণনা করে
2. **SUM()** - যোগফল বের করে
3. **AVG()** - গড় বের করে
4. **MIN()** - সর্বনিম্ন মান খুঁজে
5. **MAX()** - সর্বোচ্চ মান খুঁজে

**বিশেষ Aggregate Functions:**

6. **GROUP\_CONCAT()** - String একসাথে জোড়া দেয়
7. **COUNT(DISTINCT)** - Unique values গণনা করে
8. **VARIANCE()** - বিচ্ছুতি বের করে

## 9. **STDDEV()** - Standard deviation বের করে

---

### 1. COUNT() - গণনা করা

**COUNT(\*)** - সব rows গণনা

কোড:

```
SELECT COUNT(*) AS total_teachers  
FROM teacher2;
```

ব্যাখ্যা:

- **COUNT(\*)** টেবিলের সব rows গণনা করে
- NULL values-কে count হয়
- সবচেয়ে বেশি ব্যবহৃত aggregate function
- Result: শুধু একটা সংখ্যা

Calculation:

Row 1: Azad ✓

Row 2: Kamal ✓

Row 3: Hasan ✓

Row 4: Kamal ✓

Row 5: Rony ✓

Row 6: Rakib ✓

---

Total = 6

আউটপুট:

**total\_teachers**

**মনে রাখার কথা:** COUNT(\*) = সব rows গণনা (NULL-সহ)

---

### COUNT(column\_name) - Specific column গণনা

**কোড:**

```
SELECT COUNT(T_phone) AS teachers_with_phone  
FROM teacher2;
```

**ধরি T\_phone column আছে যেখানে কিছু NULL:**

**T\_id T\_name T\_phone**

201 Azad 0171111111

202 Kamal NULL

203 Hasan 0172222222

204 Kamal NULL

205 Rony 0173333333

206 Rakib 0174444444

**ব্যাখ্যা:**

- **COUNT(column)** শৃঙ্খু **NOT NULL values** গণনা করে
- **NULL values skip** করে
- Specific data আছে কিনা check করতে useful

**Calculation:**

Azad: 0171111111 ✓

Kamal: NULL X

Hasan: 0172222222 ✓

Kamal: NULL X

Rony: 0173333333 ✓

Rakib: 01744444444 ✓

---

Total = 4

**আউটপুট:**

**teachers\_with\_phone**

4

**মনে রাখার কথা:** COUNT(column) = শব্দ NOT NULL গণনা

---

**COUNT(DISTINCT column) - Unique values গণনা**

**কোড:**

```
SELECT COUNT(DISTINCT T_dept) AS unique_departments  
FROM teacher2;
```

**ব্যাখ্যা:**

- **COUNT(DISTINCT)** শব্দ **unique/আলাদা values** গণনা করে
- Duplicate values একবার count হয়
- কিন্তুগুলো different values আছে জানতে

**Departments List:**

Row 1: CSE ✓ (প্রথমবার)

Row 2: EEE ✓ (প্রথমবার)

Row 3: CE ✓ (প্রথমবার)

Row 4: EEE X (আগেই আছে, skip)

Row 5: CSE X (আগেই আছে, skip)

Row 6: CSE X (আগেই আছে, skip)

---

Unique = 3 (CSE, EEE, CE)

**আউটপুট:**

**unique\_departments**

3

**মনে রাখার কথা:** COUNT(DISTINCT) = শুধু unique values

---

**COUNT() with GROUP BY**

**কোড:**

```
SELECT T_dept, COUNT(*) AS teachers_per_dept  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- **GROUP BY** দিয়ে প্রতি group-এর জন্য আলাদা count
- প্রতি department-এ কতজন teacher

**Grouping:**

CSE Group:

- Azad, Rony, Rakib = 3 জন

EEE Group:

- Kamal, Kamal = 2 জন

CE Group:

- Hasan = 1 জন

**আউটপুট:**

## T\_dept teachers\_per\_dept

CSE 3

EEE 2

CE 1

মনে রাখার কথা: GROUP BY দিয়ে প্রতি group-এর count

---

## 2. SUM() - যোগফল

### Basic SUM

কোড:

```
SELECT SUM(T_salary) AS total_salary  
FROM teacher2;
```

ব্যাখ্যা:

- **SUM()** সব values যোগ করে
- শুধু **numeric columns**-এ কাজ করে
- NULL values skip করে (0 হিসেবে ধরে না)
- সব কিছু মিলিয়ে মোট করে

Calculation:

1000 (Azad)

+ 2000 (Kamal)

+ 1000 (Hasan)

+ 1000 (Kamal)

+ 2000 (Rony)

+ 4000 (Rakib)

-----  
= 11000

**আউটপুট:**

**total\_salary**

11000

**মনে রাখার কথা:**  $\text{SUM}()$  = সব মিলিয়ে যোগফল

---

**SUM() with GROUP BY**

**কোড:**

```
SELECT T_dept, SUM(T_salary) AS dept_total_salary  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- প্রতি department-এর আলাদা আলাদা মোট বেতন
- প্রতি group-এর জন্য SUM calculate হয়

**Calculation:**

CSE Department:

$$1000 + 2000 + 4000 = 7000$$

EEE Department:

$$2000 + 1000 = 3000$$

CE Department:

$$1000 = 1000$$

**আউটপুট:**

### **T\_dept dept\_total\_salary**

CSE 7000

EEE 3000

CE 1000

**মনে রাখার কথা:** GROUP BY দিয়ে প্রতি group-এর যোগফল

---

### **SUM() with Condition (CASE WHEN)**

**কোড:**

SELECT

```
SUM(CASE WHEN T_salary > 1500 THEN T_salary ELSE 0 END) AS high_salary_total,  
SUM(CASE WHEN T_salary <= 1500 THEN T_salary ELSE 0 END) AS low_salary_total  
FROM teacher2;
```

**ব্যাখ্যা:**

- **CASE WHEN** দিয়ে শর্তসাপেক্ষে যোগ করা
- High salary (>1500) এবং Low salary ( $\leq 1500$ ) আলাদা করে total

**Calculation:**

High Salary (>1500):

Kamal: 2000 ✓

Rony: 2000 ✓

Rakib: 4000 ✓

Total = 8000

Low Salary ( $\leq 1500$ ):

Azad: 1000 ✓

Hasan: 1000 ✓

Kamal: 1000 ✓

Total = 3000

**আউটপুট:**

**high\_salary\_total low\_salary\_total**

8000                  3000

**মনে রাখার কথা:** CASE WHEN দিয়ে conditional SUM

---

### 3. AVG() - গড়

#### Basic AVG

**কোড:**

```
SELECT AVG(T_salary) AS average_salary  
FROM teacher2;
```

**ব্যাখ্যা:**

- **AVG()** গড় (average) বের করে
- Formula: মোট যোগফল ÷ সংখ্যা
- NULL values skip করে
- Decimal/দশমিক আকারে result আসে

**Calculation:**

Total Salary = 1000 + 2000 + 1000 + 1000 + 2000 + 4000 = 11000

Count = 6

Average =  $11000 \div 6 = 1833.33$

**আউটপুট:**

**average\_salary**

1833.33

**মনে রাখার কথা:**  $\text{AVG}() = \text{মোট} \div \text{সংখ্যা}$

---

**AVG() with GROUP BY**

**কোড:**

```
SELECT T_dept, AVG(T_salary) AS avg_dept_salary  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- প্রতি department-এর আলাদা গড় বেতন
- কোন department-এ গড়ে কত বেতন

**Calculation:**

CSE:  $(1000 + 2000 + 4000) \div 3 = 7000 \div 3 = 2333.33$

EEE:  $(2000 + 1000) \div 2 = 3000 \div 2 = 1500.00$

CE:  $1000 \div 1 = 1000.00$

**আউটপুট:**

**T\_dept avg\_dept\_salary**

CSE 2333.33

EEE 1500.00

CE 1000.00

**মনে রাখার কথা:** GROUP BY দিয়ে প্রতি group-এর গড়

---

**AVG() with ROUND (দশমিক কমানো)**

**কোড:**

```
SELECT T_dept, ROUND(AVG(T_salary), 2) AS avg_salary  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- **ROUND()** দশমিক সংখ্যা round করে
- $\text{ROUND}(\text{value}, 2) = 2$  দশমিক পর্যন্ত
- সুন্দর করে দেখানোর জন্য

**আউটপুট:**

**T\_dept avg\_salary**

CSE	2333.33
EEE	1500.00
CE	1000.00

**মনে রাখার কথা:** ROUND() দিয়ে দশমিক নিয়ন্ত্রণ

---

#### 4. MIN() - সর্বনিম্ন

**Basic MIN**

**কোড:**

```
SELECT MIN(T_salary) AS minimum_salary  
FROM teacher2;
```

**ব্যাখ্যা:**

- **MIN()** সবচেয়ে ছোট/কম মান খুঁজে বের করে
- NULL values skip করে
- সংখ্যা, তারিখ, text সব কিছুতে কাজ করে

### Finding MIN:

Salaries: 1000, 2000, 1000, 1000, 2000, 4000

সবচেয়ে কম = 1000

### আউটপুট:

**minimum\_salary**

1000

মনে রাখার কথা:  $\text{MIN}()$  = সবচেয়ে ছোট মান

---

### MIN() with GROUP BY

#### কোড:

```
SELECT T_dept, MIN(T_salary) AS min_dept_salary  
FROM teacher2  
GROUP BY T_dept;
```

#### ব্যাখ্যা:

- প্রতি department-এর সবচেয়ে কম বেতন
- কোন department-এ minimum কত

### Finding MIN per Group:

CSE:  $\text{MIN}(1000, 2000, 4000) = 1000$

EEE:  $\text{MIN}(2000, 1000) = 1000$

CE:  $\text{MIN}(1000) = 1000$

### আউটপুট:

**T\_dept min\_dept\_salary**

CSE 1000

### **T\_dept min\_dept\_salary**

EEE 1000

CE 1000

**মনে রাখার কথা:** প্রতি group-এ সবচেয়ে ছোট

---

### **MIN() with String/Text**

**কোড:**

```
SELECT MIN(T_name) AS first_alphabetically  
FROM teacher2;
```

**ব্যাখ্যা:**

- Text/String-এও MIN কাজ করে
- **Alphabetically** প্রথম নাম (A দিয়ে শুরু)
- Dictionary order অনুযায়ী

**Alphabetical Order:**

Names: Azad, Kamal, Hasan, Kamal, Rony, Rakib

Alphabetically:

A... → Azad ✓ (প্রথম)

H... → Hasan

K... → Kamal

R... → Rakib, Rony

**আউটপুট:**

**first\_alphabetically**

Azad

**মনে রাখার কথা:** MIN() text-এ alphabetically প্রথম

---

## 5. MAX() - সর্বোচ্চ

### Basic MAX

**কোড:**

```
SELECT MAX(T_salary) AS maximum_salary  
FROM teacher2;
```

**ব্যাখ্যা:**

- **MAX()** সবচেয়ে বড় মান খুঁজে বের করে
- NULL values skip করে
- Highest value return করে

### Finding MAX:

Salaries: 1000, 2000, 1000, 1000, 2000, 4000

সবচেয়ে বেশি = 4000

**আউটপুট:**

**maximum\_salary**

4000

**মনে রাখার কথা:** MAX() = সবচেয়ে বড় মান

---

### MAX() with GROUP BY

**কোড:**

```
SELECT T_dept, MAX(T_salary) AS max_dept_salary  
FROM teacher2
```

GROUP BY T\_dept;

ব্যাখ্যা:

- প্রতি department-এর সবচেয়ে বেশি বেতন
- কোন department-এ maximum কত

**Finding MAX per Group:**

CSE: MAX(1000, 2000, 4000) = 4000

EEE: MAX(2000, 1000) = 2000

CE: MAX(1000) = 1000

আউটপুট:

**T\_dept max\_dept\_salary**

CSE 4000

EEE 2000

CE 1000

মনে রাখার কথা: প্রতি group-এ সবচেয়ে বড়

---

**MAX() with String/Text**

কোড:

```
SELECT MAX(T_name) AS last_alphabetically  
FROM teacher2;
```

ব্যাখ্যা:

- Text-এ MAX মানে alphabetically শেষ নাম
- Z দিকে যেটা

**Alphabetical Order (Reverse):**

Names: Azad, Kamal, Hasan, Kamal, Rony, Rakib

Reverse Alphabetically:

R... → Rony ✓ (শেষ)

R... → Rakib

K... → Kamal

H... → Hasan

A... → Azad

**আউটপুট:**

**last\_alphabetically**

Rony

মনে রাখার কথা: MAX() text-এ alphabetically শেষ

---

## 6. GROUP\_CONCAT() - String জোড়া দেওয়া

**Basic GROUP\_CONCAT**

**কোড:**

```
SELECT GROUP_CONCAT(T_name) AS all_teacher_names  
FROM teacher2;
```

**ব্যাখ্যা:**

- **GROUP\_CONCAT()** সব values একসাথে জোড়া দেয় (concatenate)
- Default separator: কমা (,)
- সব নাম এক string-এ

**Concatenation:**

Azad, Kamal, Hasan, Kamal, Rony, Rakib

**আউটপুট:**

### **all\_teacher\_names**

Azad,Kamal,Hasan,Kamal,Rony,Rakib

**মনে রাখার কথা:** GROUP\_CONCAT = সব একসাথে জোড়

---

### **GROUP\_CONCAT with GROUP BY**

**কোড:**

```
SELECT T_dept, GROUP_CONCAT(T_name) AS teachers  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- প্রতি department-এর সব teachers-এর নাম একসাথে
- কোন department-এ কারা কারা আছে

**Grouping:**

CSE: Azad, Rony, Rakib

EEE: Kamal, Kamal

CE: Hasan

**আউটপুট:**

#### **T\_dept teachers**

CSE Azad,Rony,Rakib

EEE Kamal,Kamal

CE Hasan

**মনে রাখার কথা:** প্রতি group-এর items একসাথে

---

### **GROUP\_CONCAT with Custom Separator**

**কোড:**

```
SELECT T_dept,  
       GROUP_CONCAT(T_name SEPARATOR ' | ') AS teachers  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- **SEPARATOR** দিয়ে নিজের পছন্দের separator ব্যবহার করা যায়
- কমা (,) এর পরিবর্তে pipe (|), dash (-), space ইত্যাদি

**আউটপুট:**

**T\_dept teachers**

```
CSE      Azad | Rony | Rakib  
EEE      Kamal | Kamal  
CE       Hasan
```

**মনে রাখার কথা:** SEPARATOR দিয়ে custom separator

---

**GROUP\_CONCAT with ORDER BY**

**কোড:**

```
SELECT T_dept,  
       GROUP_CONCAT(T_name ORDER BY T_salary DESC SEPARATOR ',') AS teachers_by_salary  
FROM teacher2  
GROUP BY T_dept;
```

**ব্যাখ্যা:**

- **ORDER BY** দিয়ে concatenate করার আগে সাজানো যায়
- বেতন অনুযায়ী বড় থেকে ছোট সাজিয়ে নাম জোড়া

### **Sorting & Concatenation:**

CSE (sorted by salary DESC):

Rakib (4000), Rony (2000), Azad (1000)

→ "Rakib, Rony, Azad"

EEE (sorted by salary DESC):

Kamal (2000), Kamal (1000)

→ "Kamal, Kamal"

### **আউটপুট:**

#### **T\_dept teachers\_by\_salary**

CSE Rakib, Rony, Azad

EEE Kamal, Kamal

CE Hasan

**মনে রাখার কথা:** ORDER BY দিয়ে sorted concatenation

---

### **GROUP\_CONCAT with DISTINCT**

#### **কোড:**

```
SELECT T_dept,  
       GROUP_CONCAT(DISTINCT T_name) AS unique_teachers  
FROM teacher2  
GROUP BY T_dept;
```

#### **ব্যাখ্যা:**

- **DISTINCT** দিয়ে duplicate names একবার নেয়
- EEE-তে Kamal দুইবার আছে, কিন্তু একবার দেখাবে

### **Removing Duplicates:**

EEE: Kamal, Kamal → Kamal (duplicate removed)

আউটপুট:

### T\_dept unique\_teachers

CSE Azad,Rony,Rakib

EEE Kamal

CE Hasan

মনে রাখার কথা: DISTINCT = duplicate বাদ

---

## 7. Multiple Aggregates একসাথে

All Statistics at Once

কোড:

SELECT

```
T_dept,  
COUNT(*) AS total_teachers,  
SUM(T_salary) AS total_salary,  
AVG(T_salary) AS avg_salary,  
MIN(T_salary) AS min_salary,  
MAX(T_salary) AS max_salary,  
MAX(T_salary) - MIN(T_salary) AS salary_range
```

FROM teacher2

GROUP BY T\_dept;

ব্যাখ্যা:

- একাধিক aggregate functions একসাথে ব্যবহার করা যায়
- Complete statistics এক query-তে
- salary\_range** = সর্বোচ্চ - সর্বনিম্ন (বেতনের পার্থক্য)

### **Calculation for CSE:**

Total teachers: 3

Total salary:  $1000 + 2000 + 4000 = 7000$

Average:  $7000 \div 3 = 2333.33$

Minimum: 1000

Maximum: 4000

Range:  $4000 - 1000 = 3000$

### **আউটপুট:**

**T\_dept total\_teachers total\_salary avg\_salary min\_salary max\_salary salary\_range**

CSE	3	7000	2333.33	1000	4000	3000
EEE	2	3000	1500.00	1000	2000	1000
CE	1	1000	1000.00	1000	1000	0

**মনে রাখার কথা:** Multiple aggregates = complete analysis

---

## **8. Aggregate Functions with NULL**

### **COUNT with NULL**

ধরি টেবিলে NULL আছে:

**T\_id T\_name T\_salary**

201 Azad 1000

202 Kamal NULL

203 Hasan 2000

### **কোড:**

SELECT

COUNT(\*) AS total\_rows,

```
COUNT(T_salary) AS rows_with_salary,  
COUNT(*) - COUNT(T_salary) AS rows_without_salary  
FROM teacher2;
```

**Calculation:**

$\text{COUNT(*)} = 3$  (সব rows, NULL-সহ)

$\text{COUNT(T\_salary)} = 2$  (শুধু NOT NULL)

Difference =  $3 - 2 = 1$  (NULL আছে এমন)

**আউটপুট:**

**total\_rows rows\_with\_salary rows\_without\_salary**

3	2	1
---	---	---

**মনে রাখার কথা:**  $\text{COUNT(*)} \neq \text{COUNT(column)}$  যখন NULL থাকে

---

**SUM/AVG with NULL**

**কোড:**

```
SELECT  
    SUM(T_salary) AS total,  
    AVG(T_salary) AS average  
FROM teacher2;
```

**ব্যাখ্যা:**

- **SUM()** NULL skip করে, শুধু NOT NULL যোগ করে
- **AVG()** NULL skip করে গণনা করে
- NULL কে 0 হিসেবে ধরে না

**Calculation:**

Salaries: 1000, NULL, 2000

SUM:  $1000 + 2000 = 3000$  (NULL skip)

AVG:  $3000 \div 2 = 1500$  (2টা NOT NULL value)

**আউটপুট:**

**total average**

3000 1500

**মনে রাখার কথা:** Aggregates NULL skip করে

---

### **COALESCE with Aggregates**

**কোড:**

```
SELECT  
    T_name,  
    COALESCE(T_salary, 0) AS salary  
FROM teacher2;
```

**ব্যাখ্যা:**

- **COALESCE()** NULL-কে অন্য মান দিয়ে replace করে
- NULL থাকলে 0 দেখাবে
- Aggregates-এর আগে NULL handle করতে useful

**আউটপুট:**

**T\_name salary**

Azad 1000

Kamal 0

Hasan 2000

**মনে রাখার কথা:** COALESCE = NULL replacement

---

## 9. Advanced Aggregate Examples

### Percentage Calculation

কোড:

```
SELECT  
    T_dept,  
    COUNT(*) AS dept_count,  
    (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM teacher2)) AS percentage  
FROM teacher2  
GROUP BY T_dept;
```

ব্যাখ্যা:

- প্রতি department-এ কত শতাংশ teacher আছে
- Subquery দিয়ে মোট সংখ্যা নেওয়া
- 100.0 (decimal) ব্যবহার করা জন্মে

Calculation:

Total teachers = 6

CSE:  $(3/6) \times 100 = 50\%$

EEE:  $(2/6) \times 100 = 33.33\%$

CE:  $(1/6) \times 100 = 16.67\%$

আউটপুট:

**T\_dept dept\_count percentage**

CSE	3	50.00
EEE	2	33.33
CE	1	16.67

মনে রাখার কথা: Subquery দিয়ে percentage

---

## Cumulative Sum (Running Total)

কোড:

```
SELECT  
    T_id,  
    T_name,  
    T_salary,  
    SUM(T_salary) OVER (ORDER BY T_id) AS running_total  
FROM teacher2  
ORDER BY T_id;
```

ব্যাখ্যা:

- **Window Function** (OVER clause)
- প্রতি row-এ এ পর্যন্ত মোট কত