

## SQL HAVING

**SQL HAVING - বিস্তারিত ব্যাখ্যা (বাংলায়)**

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

---

**HAVING কী?**

HAVING হলো GROUP BY-এর পরে **groups filter** করার জন্য ব্যবহৃত clause।

**WHERE vs HAVING - মূল পার্থক্য:**

<b>বিষয়</b>	<b>WHERE</b>	<b>HAVING</b>
--------------	--------------	---------------

কখন কাজ করে	GROUP BY-এর আগে	GROUP BY-এর পরে
-------------	-----------------	-----------------

কী filter করে	Individual rows	Groups
---------------	-----------------	--------

Aggregate function **X** ব্যবহার করা যায় না **✓** ব্যবহার করা যায়

উদাহরণ WHERE T\_salary > 1000 HAVING SUM(T\_salary) > 2000

---

**1. Basic HAVING with COUNT (গণনা করে filter)**

**কোড:**

```

SELECT T_dept, COUNT(*) AS total_teachers
FROM teacher2
GROUP BY T_dept
HAVING COUNT(*) > 1;

```

### ব্যাখ্যা:

- প্রথমে **GROUP BY** department অনুযায়ী group করে
- প্রতি department-এ কতজন teacher আছে **COUNT** করে
- তারপর **HAVING** দিয়ে শুধু সেই departments নেয় যেখানে 1-এর বেশি teacher আছে
- একক teacher থাকা department (CE) বাদ যাবে

### Step-by-Step Process:

#### Step 1: GROUP BY করার পর সব groups:

T\_dept total\_teachers

CSE      3 ✓

EEE      2 ✓

CE        1 ✗

#### Step 2: HAVING COUNT(\*) > 1 apply করার পর:

CSE: 3 > 1 ✓ (রাখো)

EEE: 2 > 1 ✓ (রাখো)

CE: 1 > 1 ✗ (বাদ দাও)

### আউটপুট:

T\_dept total\_teachers

CSE      3

EEE      2

**মনে রাখার কথা:** HAVING COUNT(\*) > 1 = একাধিক entry আছে এমন groups

---

## 2. HAVING with SUM (যোগফল দিয়ে filter)

**কোড:**

```
SELECT T_dept, SUM(T_salary) AS total_salary  
FROM teacher2  
GROUP BY T_dept  
HAVING SUM(T_salary) > 2000;
```

**ব্যাখ্যা:**

- প্রতি department-এর মোট বেতন বের করা হচ্ছে
- শুধু সেই departments দেখাবে যেখানে মোট বেতন 2000-এর বেশি
- CE department-এর মোট বেতন 1000, তাই সেটা বাদ যাবে

**Step-by-Step Process:**

**Step 1: GROUP BY + SUM করার পর:**

**T\_dept total\_salary Check**

CSE	7000	7000 > 2000	✓
EEE	3000	3000 > 2000	✓
CE	1000	1000 > 2000	✗

**Calculation:**

CSE:  $1000 + 2000 + 4000 = 7000$  ✓

EEE:  $2000 + 1000 = 3000$  ✓

CE:  $1000 = 1000$  ✗

**আউটপুট:**

### **T\_dept total\_salary**

CSE 7000

EEE 3000

**মনে রাখার কথা:** HAVING SUM() > value = মোট যোগফল বেশি এমন groups

---

### **3. HAVING with AVG (গড় দিয়ে filter)**

**কোড:**

```
SELECT T_dept, AVG(T_salary) AS avg_salary
FROM teacher2
GROUP BY T_dept
HAVING AVG(T_salary) > 1500;
```

**ব্যাখ্যা:**

- প্রতি department-এর গড় বেতন বের করা হচ্ছে
- শুধু সেই departments যেখানে গড় বেতন 1500-এর বেশি
- EEE এবং CE-র গড় বেতন 1500-এর কম, তাই বাদ যাবে

**Step-by-Step Process:**

**Step 1: GROUP BY + AVG করার পর:**

#### **T\_dept avg\_salary Check**

CSE 2333.33 2333.33 > 1500 ✓

EEE 1500.00 1500 > 1500 ✗

CE 1000.00 1000 > 1500 ✗

**Calculation:**

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

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

CE:  $1000 \div 1 = 1000.00$  ✗

আউটপুট:

T\_dept avg\_salary

CSE 2333.33

মনে রাখার কথা: HAVING AVG() > value = গড় বেশি এমন groups

---

#### 4. HAVING with MIN (সর্বনিম্ন দিয়ে filter)

কোড:

```
SELECT T_dept, MIN(T_salary) AS min_salary
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING MIN(T_salary) >= 1000;
```

ব্যাখ্যা:

- প্রতি department-এর সবচেয়ে কম বেতন বের করা হচ্ছে
- শুধু সেই departments যেখানে সর্বনিম্ন বেতন 1000 বা তার বেশি
- এই উদাহরণে সব departments-ই qualify করবে কারণ সবার min = 1000

Step-by-Step Process:

Step 1: GROUP BY + MIN করার পর:

T\_dept min\_salary Check

CSE 1000 1000 >= 1000 ✓

EEE 1000 1000 >= 1000 ✓

CE 1000 1000 >= 1000 ✓

Finding MIN:

CSE: MIN(1000, 2000, 4000) = 1000 ✓

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

CE:  $\text{MIN}(1000) = 1000$  ✓

আউটপুট:

T\_dept min\_salary

CSE 1000

EEE 1000

CE 1000

মনে রাখার কথা: HAVING MIN()  $\geq$  value = সর্বনিম্ন মান check করা

---

## 5. HAVING with MAX (সর্বোচ্চ দিয়ে filter)

কোড:

```
SELECT T_dept, MAX(T_salary) AS max_salary  
FROM teacher2  
GROUP BY T_dept  
HAVING MAX(T_salary) > 2000;
```

ব্যাখ্যা:

- প্রতি department-এর সবচেয়ে বেশি বেতন বের করা হচ্ছে
- শুধু সেই departments যেখানে সর্বোচ্চ বেতন 2000-এর বেশি
- শুধু CSE qualify করবে ( $\max = 4000$ )

Step-by-Step Process:

Step 1: GROUP BY + MAX করার পর:

T\_dept max\_salary Check

CSE 4000 4000 > 2000 ✓

### T\_dept max\_salary Check

EEE 2000       $2000 > 2000$  ✗

CE 1000       $1000 > 2000$  ✗

#### Finding MAX:

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

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

CE:  $\text{MAX}(1000) = 1000$  ✗

#### আউটপুট:

### T\_dept max\_salary

CSE 4000

মনে রাখার কথা: HAVING MAX() > value = সর্বোচ্চ মান check করা

---

## 6. HAVING with Multiple Conditions (একাধিক শর্ত)

#### কোড:

```
SELECT T_dept, COUNT(*) AS total, SUM(T_salary) AS total_salary  
FROM teacher2  
GROUP BY T_dept  
HAVING COUNT(*) > 1 AND SUM(T_salary) > 3000;
```

#### ব্যাখ্যা:

- দুটি শর্ত একসাথে check করা হচ্ছে
- শর্ত 1: একাধিক teacher আছে ( COUNT > 1 )
- শর্ত 2: মোট বেতন 3000-এর বেশি ( SUM > 3000 )
- উভয় শর্ত পূরণ করতে হবে ( AND )

#### Step-by-Step Process:

**Step 1: GROUP BY + aggregates করার পর:**

T_dept	total	total_salary	Check COUNT	Check SUM	Final
CSE	3	7000	3 > 1 <span style="color: green;">✓</span>	7000 > 3000 <span style="color: green;">✓</span> <span style="color: green;">✓</span>	
EEE	2	3000	2 > 1 <span style="color: green;">✓</span>	3000 > 3000 <span style="color: red;">✗</span> <span style="color: red;">✗</span>	
CE	1	1000	1 > 1 <span style="color: red;">✗</span>	1000 > 3000 <span style="color: red;">✗</span> <span style="color: red;">✗</span>	

**Logic:**

CSE:  $(3 > 1) \text{ AND } (7000 > 3000) = \text{TRUE AND TRUE} = \checkmark$

EEE:  $(2 > 1) \text{ AND } (3000 > 3000) = \text{TRUE AND FALSE} = \times$

CE:  $(1 > 1) \text{ AND } (1000 > 3000) = \text{FALSE AND FALSE} = \times$

**আউটপুট:**

**T\_dept total total\_salary**

CSE 3 7000

**মনে রাখার কথা:** AND = সব শর্ত পূরণ করতে হবে

---

## 7. HAVING with OR Condition (যেকোনো একটি শর্ত)

**কোড:**

```
SELECT T_dept, COUNT(*) AS total, MAX(T_salary) AS max_salary  
FROM teacher2  
GROUP BY T_dept  
HAVING COUNT(*) > 2 OR MAX(T_salary) > 2000;
```

**ব্যাখ্যা:**

- দুটি শর্তের যেকোনো একটি পূরণ হলেই হবে
- শর্ত 1: 2-এর বেশি teacher (COUNT > 2)
- শর্ত 2: সর্বোচ্চ বেতন 2000-এর বেশি (MAX > 2000)

- যেকোনো একটি TRUE হলেই আসবে ( OR)

#### **Step-by-Step Process:**

##### **Step 1: GROUP BY + aggregates করার পর:**

	T_dept	total	max_salary	Check COUNT	Check MAX	Final
CSE	3	4000		3 > 2 ✓	4000 > 2000 ✓ ✓	
EEE	2	2000		2 > 2 ✗	2000 > 2000 ✗ ✗	
CE	1	1000		1 > 2 ✗	1000 > 2000 ✗ ✗	

#### **Logic:**

CSE:  $(3 > 2) \text{ OR } (4000 > 2000) = \text{TRUE OR TRUE} = \checkmark$

EEE:  $(2 > 2) \text{ OR } (2000 > 2000) = \text{FALSE OR FALSE} = \times$

CE:  $(1 > 2) \text{ OR } (1000 > 2000) = \text{FALSE OR FALSE} = \times$

#### **আউটপুট:**

##### **T\_dept total max\_salary**

CSE     3     4000

মনে রাখার কথা: OR = যেকোনো একটি শর্ত পূরণ হলেই হবে

---

#### **8. WHERE + GROUP BY + HAVING (সব একসাথে)**

#### **কোড:**

```
SELECT T_dept, COUNT(*) AS total, AVG(T_salary) AS avg_salary
FROM teacher2
WHERE T_salary > 1000
GROUP BY T_dept
HAVING COUNT(*) > 1;
```

#### **ব্যাখ্যা:**

- **WHERE:** প্রথমে individual rows filter (বেতন 1000-এর বেশি)
- **GROUP BY:** তারপর department অনুযায়ী group
- **HAVING:** শেষে groups filter (একাধিক teacher আছে এমন)

#### **Step-by-Step Process:**

##### **Step 1: WHERE T\_salary > 1000 (rows filter):**

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

202	Kamal	EEE	2000	✓
205	Rony	CSE	2000	✓
206	Rakib	CSE	4000	✓

##### **Step 2: GROUP BY T\_dept:**

T\_dept total avg\_salary

CSE	2	3000.00
EEE	1	2000.00

##### **Step 3: HAVING COUNT(\*) > 1:**

CSE: 2 > 1 ✓ (রাখো)

EEE: 1 > 1 ✗ (বাদ দাও)

**আউটপুট:**

T\_dept total avg\_salary

CSE	2	3000.00
-----	---	---------

**মনে রাখার কথা:** WHERE (rows) → GROUP BY → HAVING (groups)

## **9. HAVING with Complex Expression**

**কোড:**

```
SELECT T_dept,
```

```

SUM(T_salary) AS total_salary,
AVG(T_salary) AS avg_salary
FROM teacher2
GROUP BY T_dept
HAVING SUM(T_salary) > AVG(T_salary) * 2;

```

### ব্যাখ্যা:

- **Complex condition:** মোট বেতন > গড় বেতনের দ্বিগুণ
- যে departments-এ মোট বেতন, গড় বেতনের ২ গুণের বেশি
- Mathematical expressions HAVING-এ ব্যবহার করা যায়

### Step-by-Step Process:

#### Step 1: Calculate aggregates:

**T\_dept total\_salary avg\_salary avg\*2 Check**

CSE	7000	2333.33	4666.66	7000 > 4666.66	✓
EEE	3000	1500.00	3000.00	3000 > 3000	✗
CE	1000	1000.00	2000.00	1000 > 2000	✗

### Logic:

CSE:  $7000 > (2333.33 * 2) \rightarrow 7000 > 4666.66$  ✓

EEE:  $3000 > (1500 * 2) \rightarrow 3000 > 3000$  ✗

CE:  $1000 > (1000 * 2) \rightarrow 1000 > 2000$  ✗

### আউটপুট:

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

CSE	7000	2333.33
-----	------	---------

**মনে রাখার কথা:** HAVING-এ mathematical expressions ব্যবহার করা যায়

---

## 10. HAVING with BETWEEN

কোড:

```
SELECT T_dept, COUNT(*) AS total  
FROM teacher2  
GROUP BY T_dept  
HAVING COUNT(*) BETWEEN 2 AND 5;
```

ব্যাখ্যা:

- **BETWEEN** একটা range check করে
- ২ থেকে ৫ জনের মধ্যে teacher আছে এমন departments
- BETWEEN inclusive (2 এবং 5 include হবে)

Step-by-Step Process:

Step 1: GROUP BY + COUNT:

**T\_dept total Check**

CSE     3     3 BETWEEN 2 AND 5 ✓

EEE     2     2 BETWEEN 2 AND 5 ✓

CE     1     1 BETWEEN 2 AND 5 ✗

Logic:

CSE:  $2 \leq 3 \leq 5$  ✓

EEE:  $2 \leq 2 \leq 5$  ✓

CE:  $2 \leq 1 \leq 5$  ✗

আউটপুট:

**T\_dept total**

CSE     3

## T\_dept total

EEE 2

মনে রাখার কথা: BETWEEN = range check (inclusive)

---

### 11. HAVING with IN

কোড:

```
SELECT T_dept, COUNT(*) AS total
FROM teacher2
GROUP BY T_dept
HAVING COUNT(*) IN (2, 3);
```

ব্যাখ্যা:

- IN specific values check করে
- ঠিক 2 অথবা 3 জন teacher আছে এমন departments
- Multiple values একসাথে check করা যায়

Step-by-Step Process:

Step 1: GROUP BY + COUNT:

#### T\_dept total Check

CSE 3 3 IN (2,3) ✓

EEE 2 2 IN (2,3) ✓

CE 1 1 IN (2,3) ✗

Logic:

CSE: 3 = 2? NO, 3 = 3? YES ✓

EEE: 2 = 2? YES ✓

CE: 1 = 2? NO, 1 = 3? NO ✗

**আউটপুট:**

**T\_dept total**

CSE      3

EEE      2

**মনে রাখার কথা:** IN = specific values check

---

## 12. HAVING with NOT

**কোড:**

```
SELECT T_dept, COUNT(*) AS total
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING NOT COUNT(*) = 1;
```

**ব্যাখ্যা:**

- NOT condition-কে উল্টে দেয়
- যে departments-এ ঠিক 1 জন teacher নেই
- মানে 1-এর বেশি teacher আছে এমন departments

**Step-by-Step Process:**

**Step 1: GROUP BY + COUNT:**

**T\_dept total COUNT=1? NOT?**

CSE      3      FALSE      TRUE

EEE      2      FALSE      TRUE

CE      1      TRUE      FALSE

**Logic:**

CSE: NOT (3 = 1) → NOT FALSE → TRUE

EEE: NOT (2 = 1) → NOT FALSE → TRUE ✓

CE: NOT (1 = 1) → NOT TRUE → FALSE ✗

আউটপুট:

T\_dept total

CSE 3

EEE 2

মনে রাখার কথা: NOT = condition উল্টানো

---

সাধারণ ভুল এবং সমাধান:

✗ ভুল 1: WHERE-এ aggregate function ব্যবহার করা

-- ✗ ভুল - Error হবে!

```
SELECT T_dept, COUNT(*)
```

```
FROM teacher2
```

```
WHERE COUNT(*) > 1
```

```
GROUP BY T_dept;
```

কেন ভুল?

- WHERE grouping-এর আগে কাজ করে
- Aggregate functions (COUNT, SUM, etc.) GROUP BY-এর পরে calculate হয়
- তাই WHERE-এ aggregate function ব্যবহার করা যায় না

✓ সঠিক:

```
SELECT T_dept, COUNT(*)
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING COUNT(*) > 1;
```

---

## **✗ ভুল 2: HAVING ছাড়া aggregate condition**

-- ✗ ভুল - HAVING লাগবে!

```
SELECT T_dept, SUM(T_salary)  
FROM teacher2  
GROUP BY T_dept  
SUM(T_salary) > 2000;
```

### **✓ সঠিক:**

```
SELECT T_dept, SUM(T_salary)  
FROM teacher2  
GROUP BY T_dept  
HAVING SUM(T_salary) > 2000;
```

---

## **✗ ভুল 3: GROUP BY ছাড়া HAVING**

-- ✗ ভুল - GROUP BY লাগবে!

```
SELECT T_dept, COUNT(*)  
FROM teacher2  
HAVING COUNT(*) > 1;
```

### **কেন ভুল?**

- HAVING সবসময় GROUP BY-এর সাথে ব্যবহার হয়
- GROUP BY ছাড়া HAVING অথইন

### **✓ সঠিক:**

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

HAVING COUNT(\*) > 1;

---

#### ✗ ভুল 4: SELECT-এ নেই এমন column HAVING-এ

-- ✗ Confusing (কাজ করবে কিন্তু confusing)

```
SELECT T_dept, COUNT(*)
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING SUM(T_salary) > 2000;
```

**সমস্যা:** SUM(T\_salary) SELECT-এ নেই, তাই user দেখতে পাবে না

#### ✓ ভালো:

```
SELECT T_dept, COUNT(*), SUM(T_salary) AS total
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING SUM(T_salary) > 2000;
```

---

#### WHERE vs HAVING - বিশ্লেষিত তুলনা:

##### Scenario 1: Individual row condition

-- ✓ WHERE ব্যবহার করুন

```
SELECT * FROM teacher2 WHERE T_salary > 1000;
```

##### Scenario 2: Group-based condition

-- ✓ HAVING ব্যবহার করুন

```
SELECT T_dept, SUM(T_salary)
```

```
FROM teacher2
```

```
GROUP BY T_dept
```

```
HAVING SUM(T_salary) > 2000;
```

### Scenario 3: উভয়ই একসাথে

--  প্রথমে WHERE, তারপর HAVING

```
SELECT T_dept, COUNT(*)  
FROM teacher2  
WHERE T_salary > 1000      -- rows filter  
GROUP BY T_dept  
HAVING COUNT(*) > 1;      -- groups filter
```

---

### Query Execution Order (মনে রাখুন):

1. FROM → table নেওয়া
2. WHERE → 🔥 individual rows filter
3. GROUP BY → grouping করা
4. HAVING → 🔥 groups filter
5. SELECT → columns select করা
6. ORDER BY → সাজানো
7. LIMIT → সংখ্যা সীমিত করা

### মনে রাখার formula:

FROM → WHERE (rows) → GROUP BY → HAVING (groups) → SELECT → ORDER BY

---

### HAVING Summary Table:

Condition Type	Syntax Example	ব্যাখ্যা
COUNT	HAVING COUNT(*) > 1	একাধিক entry
SUM	HAVING SUM(col) > 1000	মোট যোগফল

Condition Type Syntax Example		ব্যাখ্যা
<b>AVG</b>	HAVING AVG(col) > 50	গড় মান
<b>MIN</b>	HAVING MIN(col) >= 10	সর্বনিম্ন মান
<b>MAX</b>	HAVING MAX(col) < 100	সর্বোচ্চ মান
<b>AND</b>	HAVING ... AND ...	সব শর্ত
<b>OR</b>	HAVING ... OR ...	যেকোনো একটি
<b>BETWEEN</b>	HAVING COUNT(*) BETWEEN 2 AND 5 Range	
<b>IN</b>	HAVING COUNT(*) IN (2,3)	Specific values
<b>NOT</b>	HAVING NOT COUNT(*) = 1	উল্টো

---

### বাস্তব জীবনে HAVING ব্যবহার:

#### 1. Sales Analysis

-- যে products-এর মোট বিক্রয় 10000-এর বেশি

```
SELECT product_name, SUM(sales_amount)
```

```
FROM sales
```

```
GROUP BY product_name
```

```
HAVING SUM(sales_amount) > 10000;
```

#### 2. Student Performance

-- যে classes-এর গড় marks 75-এর বেশি

```
SELECT class, AVG(marks)
```

```
FROM students
```

```
GROUP BY class
```

```
HAVING AVG(marks) > 75;
```

### 3. E-commerce

-- যে customers 3-এর বেশি orders দিয়েছে

```
SELECT customer_id, COUNT(*)  
FROM orders  
GROUP BY customer_id  
HAVING COUNT(*) > 3;
```

### 4. Salary Report

-- যে departments-এর মোট খরচ বাজেট exceed করেছে

```
SELECT dept, SUM(salary)  
FROM employees  
GROUP BY dept  
HAVING SUM(salary) > 100000;
```

### 5. Website Analytics

-- যে pages-এ 1000-এর বেশি visits

```
SELECT page_url, COUNT(*)  
FROM page_views  
GROUP BY page_url  
HAVING COUNT(*) > 1000;
```

---