Class will start at 9:05 PM

Agenda

→ Aggregate queries
→ Aggregation functions
  - COUNT
  - *
→ Other aggregate functions
→ Group By
→ HAVING.

] → Imp

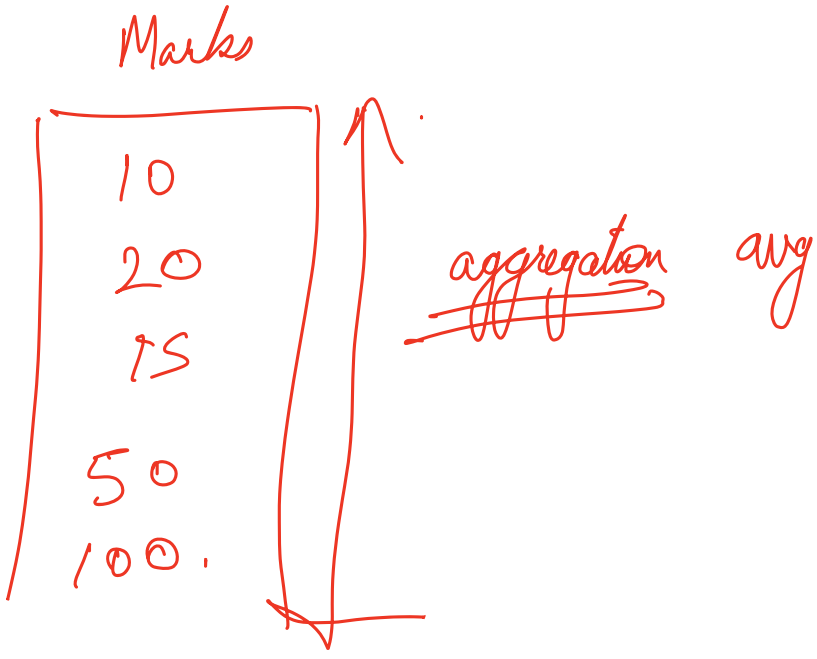
# Aggregate Queries

## Till Now

- Find students who - - - - - .
- Find the batches - - - - - - - .
- Find name of students who - - - .

## Now :→

- What are the avg mark of all students ?
- What is the average salary of all employees ?



| avg marks |
| --- |
| 39 |

Aggregate functions

count, min, max, sum, avg.

Aggregate queries are the queries that look at the value of a particular column for all rows and perform and aggregate function on them.

students

| id | name | batch_id |
| --- | --- | --- |
| 1 | A | 1 |
| 2 | B | 2 |
| 5 | C | 3 |
| 3 | D | 4 |
| 4 | E | 5 |
| 6 | F | NULL |

```
select count(id)
from students;
```

```
select count (batch_id) ] → 5
from students;
```

Note:→ Aggregate functions do not include null values.

students

| id | name | batch_id |
|---|---|---|
| 1 | A | 1 |
| 2 | B | 2 |
| 5 | C | 1 |
| 3 | D | 1 |
| 4 | E | 2 |
| 6 | F | NULL |

count (batch_id) → counts duplicates.

count (batch_id) → Ans = 5

count ( distinct batch_id) → unique

## Pseudo code

```
count = 0
for (row in table)
    if row['batch_id'] != null
        count += 1;

print count;
```

Error

```
select count(*), batch_id from students;
```

Yes

```
select count(batch_id), count(id)
from students;
```

When using aggregate queries, you cannot give any column name in select.
But you can give other aggregate function.

| count(batch_id) | count(id) |
|---|---|
| 5 | 6 |

## Other aggregate function

MIN :⇒ Min value for a col

MAX : Max value for a col

SUM : Sum of all values for a col.

AVG : Avg of all " " " "

students

| id | name | marks |
|---|---|---|
| 1 | A | 100 |
| 2 | B | 90 |
| 3 | C | 80 |
| 4 | D | 100 |
| 5 | E | 70 |
| 6 | F | NULL |

count (marks) :⇒ 5
min (marks) :⇒ 70
max (marks) :⇒ 100
avg (marks) :⇒ 88.
sum (marks) :⇒ 440

10 min break;

class resumes at 10:10 PM

## GROUP BY

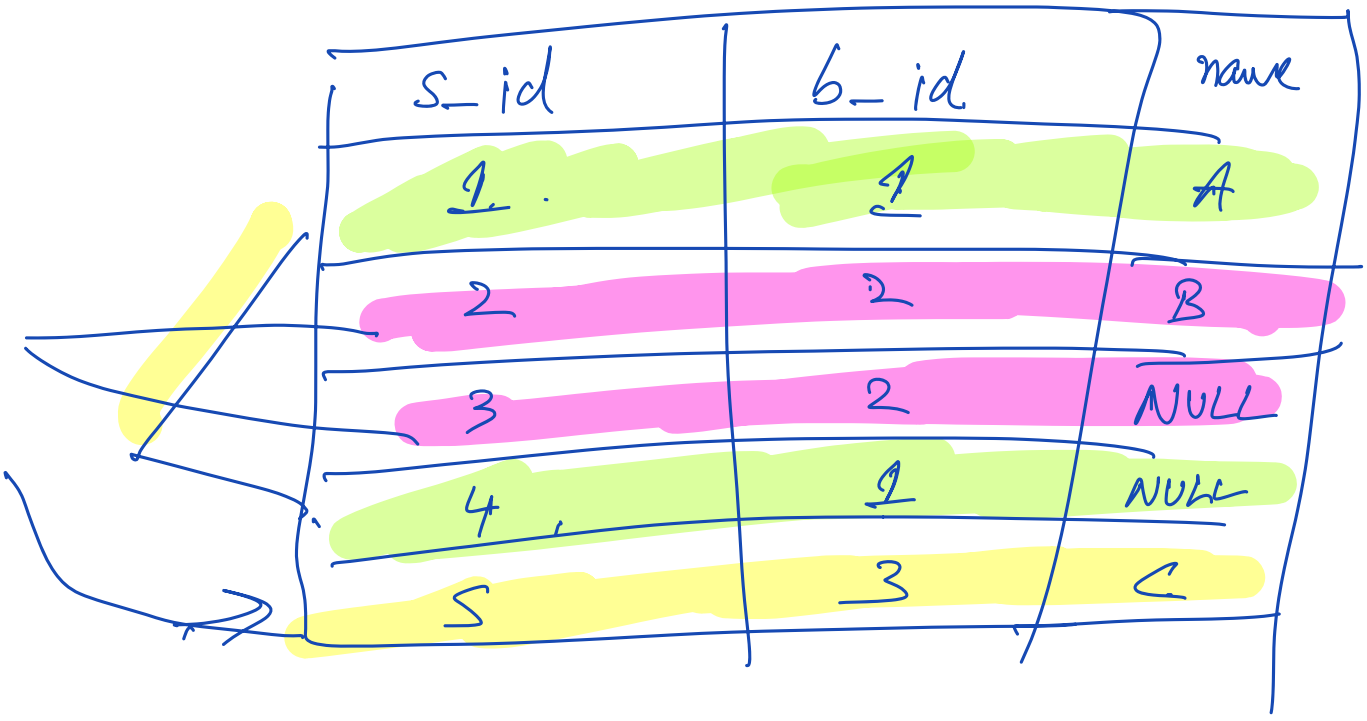
students

| id | name | b_id |
|---|---|---|
| 1 | A | 1 |
| 2 | B | 1 |
| 3 | C | 2 |
| 4 | D | 3 |
| 5 | E | 2 |

set 1
set 3
set 2

| count(*) | batch_id |
|---|---|
| 2 | 1 |
| 2 | 2 |
| 1 | 3 |

```
select count(*), b_id
from students
group by ( b_id );
```




```
select count(name), b_id.
from students
group by ( b_id);
```

| count (name) | b_id |
|---|---|
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |

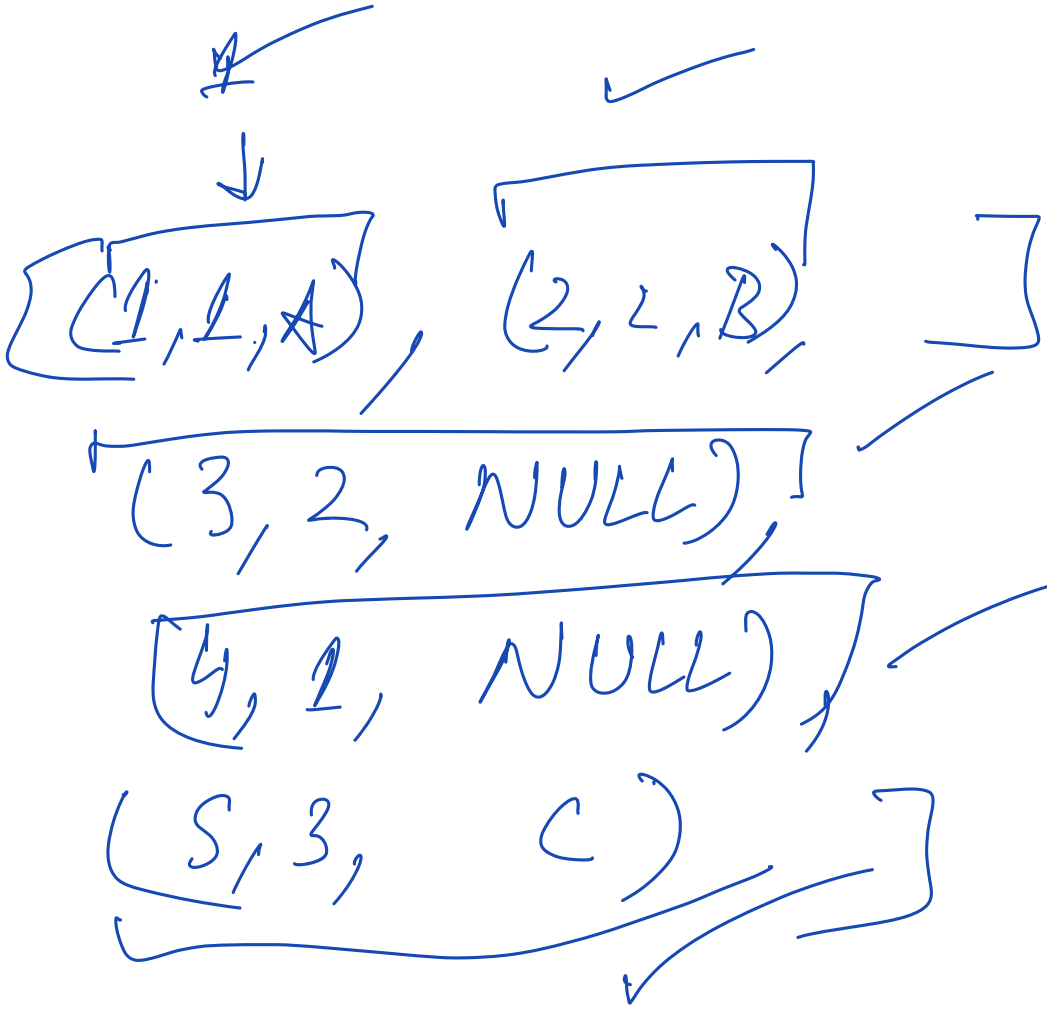
```
select count(*), b_id
from students
group by (b_id);
```

| count (*) | b_id |
|---|---|
| 2 | 1 |
| 2 | 2 |
| 1 | 3 |

count (name)

[A, NULL, B]

Ans = 2

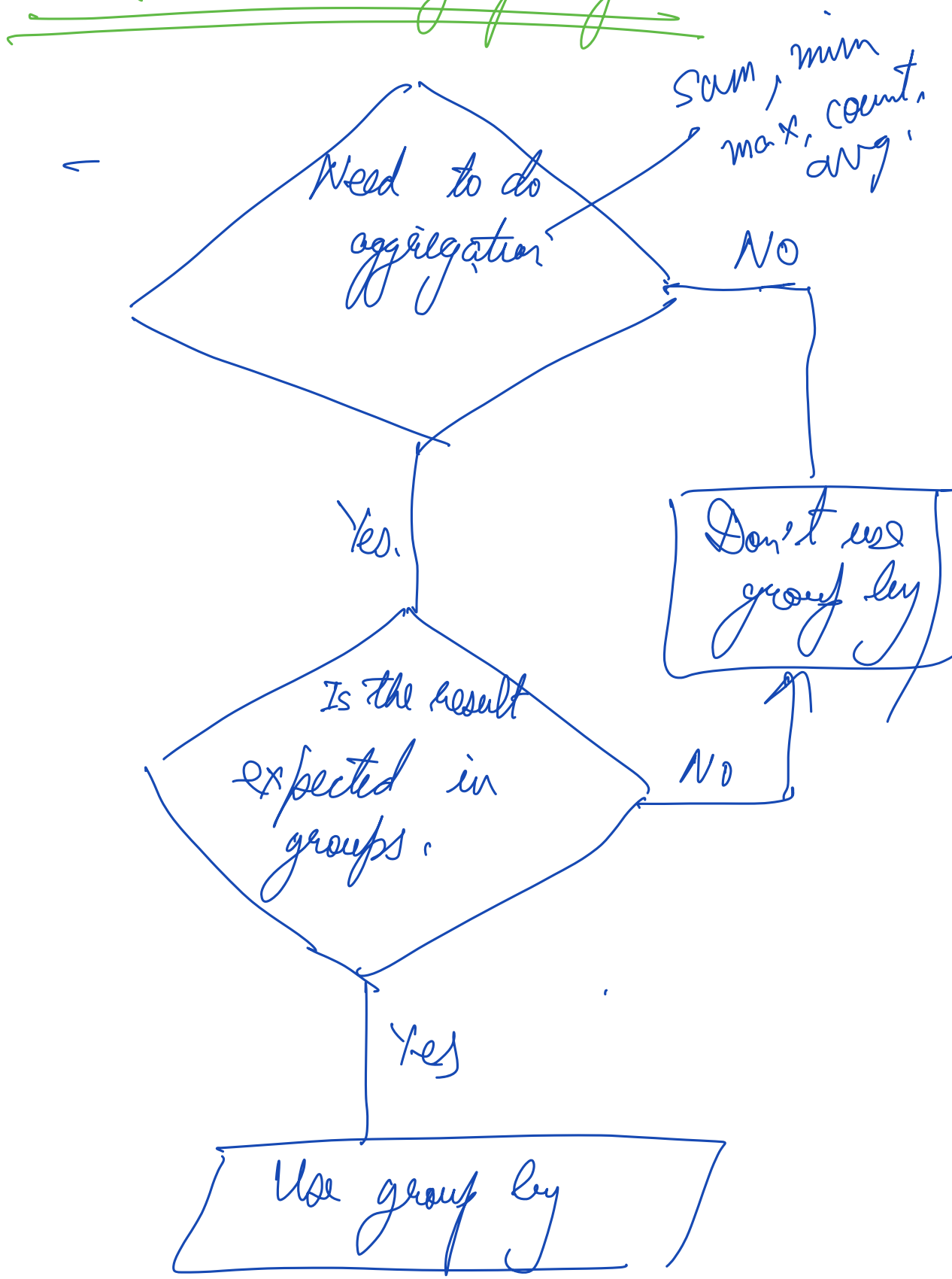



```
select count(*), b_id, name
from students
group by (b_id);
```

Note :→

In case of group by, you can give in select either an aggregate function or a col that you have used in group by.

# When to use group by?



Q) Give me count of students in batches, along with batch id, for the batches that have more than 2 students.

students

| id | name | b_id |
|---|---|---|
| 1 | A | 1 |
| 2 | B | 2 |
| 3 | C | 3 |
| 4 | D | 4 |
| 5 | E | 4 |
| 6 | F | 4 |
| 7 | G | 1 |
| 8 | H | 1 |

| count (*) | b_id |
|---|---|
| 3 | 1 |
| 3 | 4 |

```
select count(*)
from students
where count(*) > 2;
```

cannot use an aggregate function in where.

where → for row in table
if (row matches cond in where)
ans.append(row);

(1) Count (*) for all batches

| count (*) | b_id |
|---|---|
| 3 | 1 |
| 1 | 2 |
| 1 | 3 |
| 3 | 4 |

(2) Filter further only those that have count (*) > 2

| count (*) | b_id |
|---|---|
| 3 | 1 |
| 3 | 4 |

```
select count(*), b_id
from students
group by (b_id)
having count(*) > 2;
```

aggregate function or col in a group by.

## Having

If after the result of group by, you want to show only some rows from result of group by you use having.

| where | having |
|---|---|
| → before group by | → after group by. |
| → can not use aggregate function | → can use aggregate function. |

# ORDER of execution :

1. FROM
2. JOIN
3. WHERE
4. GROUP BY
5. HAVING
6. order by
7. SELECT

Students

| id | name | marks | b_id |
|---|---|---|---|
| 1 | A | 90 | 1 |
| 2 | B | 91 | 1 |
| 3 | C | 70 | 1 |
| 4 | D | 80 | 2 |
| 5 | E | 90 | 2 |
| 6 | F | 75 | 2 |
| 7 | G | 50 | 3 |
| 8 | H | 90 | 4 |
| 9 | I | 95 | 4 |

Q) Give me count of toppers in each bath.

Topper $\geq 85$

:→

```
select count(*), b_id.
from students
where marks >= 85
group by (b_id)
```

Q) Give me id of batches with more than 2 toppers;

```
Select b_id
from students
where marks ≥ 85
group by (b_id)
having count(*) ≥ 2
```