Today's Agenda :-

1) Aggregate Queries

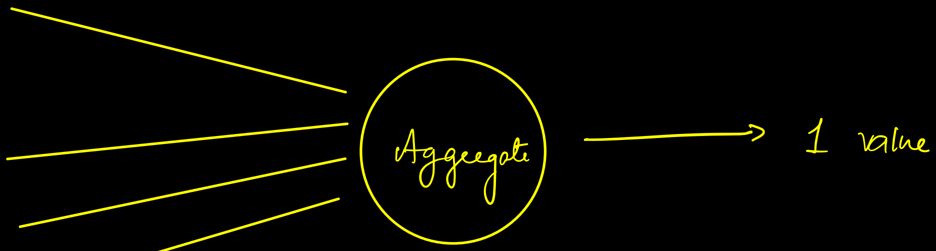
2) Aggregate Functions

3) HAVING

4) GROUP BY

---

Aggregate Function :- (COUNT, MIN, SUM, MAX, AVG,)



1) COUNT :- takes a lot of values & combines them into a single value which is equal to count of NON-NULL values in the set.

Eg :- count (1, 2, 3, 4, 5) = 5

count (1, 2) = 2

Students

| id | Name | age | b_id |
|---|---|---|---|
| 1 | A | 20 | 1 |
| 2 | B | 25 | 1 |
| 3 | C | 26 | NULL |
| 4 | D | 19 | 2 |

Q) Give count of students have a batch.

```
Select count(b_id)
from Student
```

→ 4

```
Select count(id)
from Students
```

$\boxed{\text{count}(1, 1, \text{NULL}, 2) = 3}$

↳ All aggregate function ignore NULL

Q) Give count of Students that have a batch but consider only those whose age < 23.

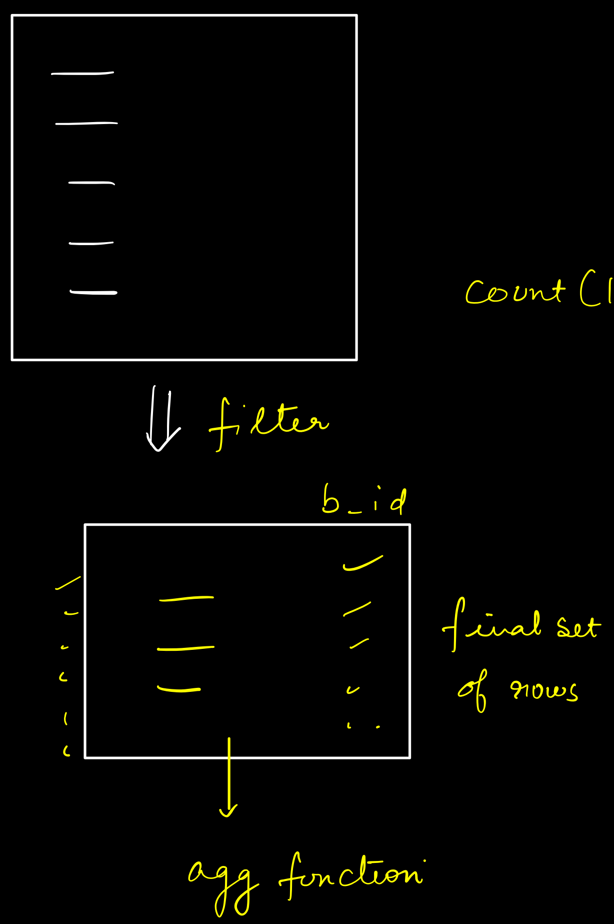
Students

| id | Name | age | b_id |
|---|---|---|---|
| 1 | A | 20 | 1 |
| 2 | B | 25 | 1 |
| 3 | C | 26 | NULL |
| 4 | D | 19 | 2 |

$\Rightarrow$

```
Select count(b_id)
from Students
where age < 23
```

count(1, 5, ~~NULL~~, ~~NULL~~) = $\underline{\underline{2}}$



Q) Get count of students with batch name = 'A'

Students

| id | Name | b. id |
|---|---|---|
| 1 | a | 1 |
| 2 | b | 1 |
| 3 | c | 2 |

Batches

| id | Name |
|---|---|
| 1 | A |
| 2 | B |

→ (1, 2) = 2

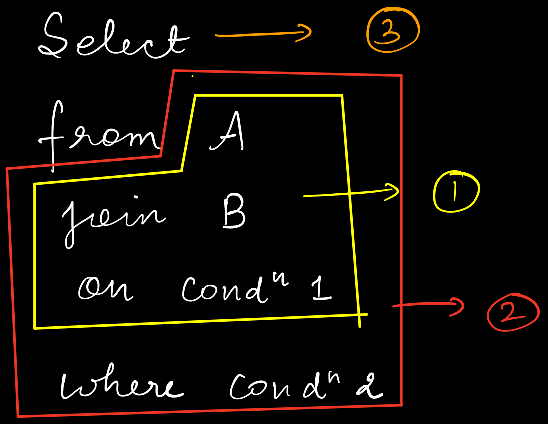
```
Select count(s.id)
from Students s
join Batches b
on s.b_id = b.id
where b.name = 'A'
```

✓

| Students | | | Batches | | |
|---|---|---|---|---|---|
| 1 | a | 1 | 1 | A | ✓ |
| 2 | b | 1 | 1 | A | ✓ |
| ~~3~~ | ~~c~~ | ~~2~~ | ~~2~~ | ~~B~~ | |

Count(*)

2



```
Select
from  A
join  B
  on  condn 1
where condn 2
```

```
A = [ ]
B = [ ]
ans1 = [ ]

for each row1 in A
    for each row2 in B
        if ( cond1 is true)
            ans1. add (row1 + row2)

ans2 = [ ]
for each row in ans1
    if (row satisfies cond2)
        ans2. add(row)
```

```
count_s_id = 0

for each row in ans2

        if( row[s_id] is not NULL)

                count_s_id ++

print (count_s_id)
```

Students

| id | Name | age | b_id |
|---|---|---|---|
| No | ~~[illegible]~~ | ~~25~~ | - - |

Q) Tell how many Students are there?

$\hookrightarrow$ Total no of row of the table

$\rightarrow$ PK

```
Select count(id)
from Students
```

id's can be NULL

```
Select Count(*)
from Students
```

✓

$\hookrightarrow$

( row[S-id], row[ag . . . )

!= NULL)

```
count_s_id = 0

for each row in ans2
    if ((row[s_id]) is not NULL)
        count_s_id ++
```

count(s_id)

1

```
Select count(1)
from Student
```

$\longrightarrow$ count of all the rows

count(*) = count(1)

$\downarrow$ * can never be NULL

$\downarrow$ 1 can never be NULL

Other Aggregate functions :-

You can print multiple aggregations at the same time

```
Select count(b_id), sum(psp), avg(psp)
 . . . .
from ( — )
```

$$\left.\begin{array}{l}\text{MAX}\\ \text{MIN}\\ \text{AVG}\end{array}\right\}$$

AVG(1, 2, 3, NULL) = avg of NON NULL values

$(1, 2, 3) \Rightarrow \frac{6}{3} = \textcircled{2}$

| id | psp |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | NULL |

Select avg(psp) ⇒ 2

!=

Select $\underline{\text{sum(psp)}}^{6}$ / $\underline{\text{count(id)}}^{4}$ ⇒ !=

---

GROUP BY

HAVING

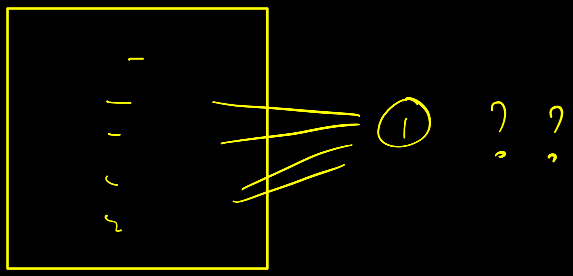
## GROUP BY :-

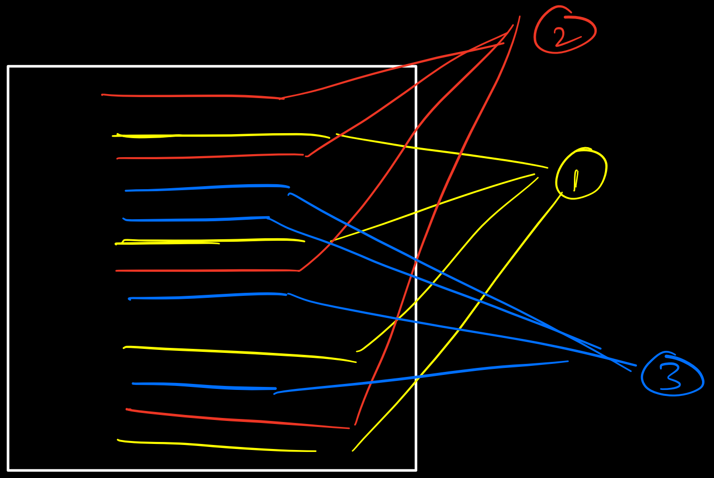
Students

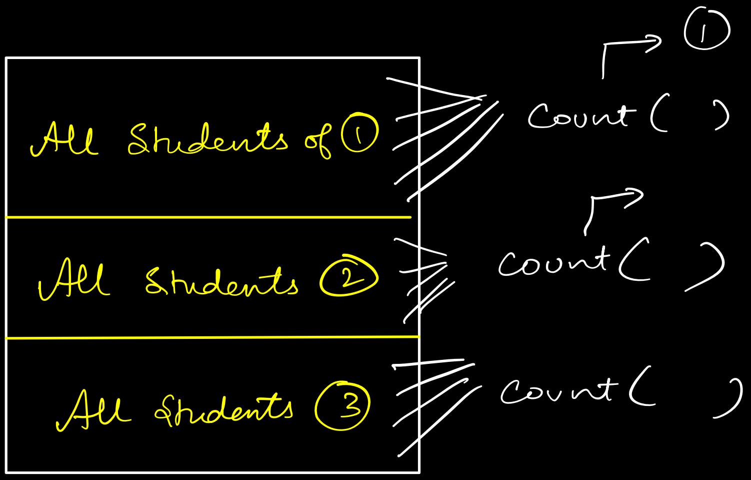
| id | Name | age | b_id |
|---|---|---|---|
| | | | 1 |
| | | | 2 |
| | | | ⋮ |

Q) Get the count of Students for every batch.

| b_id | Count |
|---|---|
| 1 | 50 |
| 2 | 20 |
| 3 | 26 |
| 4 | 50 |







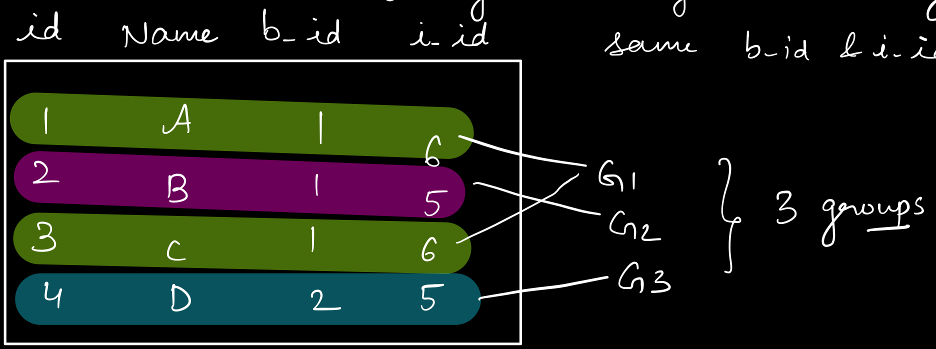
GROUP BY ⇒ allows you to break your table in multiple groups so as to be used by aggregate function.

1) group by batch_id

↳ bring all rows with same b_id together

2) group by batch_id , ins_id

↳ bring rows together having same b_id & i_id



| id | Name | b_id | i_id |
|---|---|---|---|
| 1 | A | 1 | 6 |
| 2 | B | 1 | 5 |
| 3 | C | 1 | 6 |
| 4 | D | 2 | 5 |

Q) Get the count of Students for every batch.

```
Select count(*), batch_id
from Students
group by (batch_id)
```

| id | Name | b_id | psp |
|---|---|---|---|
| 1 | A | 1 | 30 |
| 2 | B | 2 | 40 |
| 3 | C | 1 | 30 |
| 4 | D | 1 | 20 |

G1

G2

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

Output

You can use only those columns in the select that are present in group by.

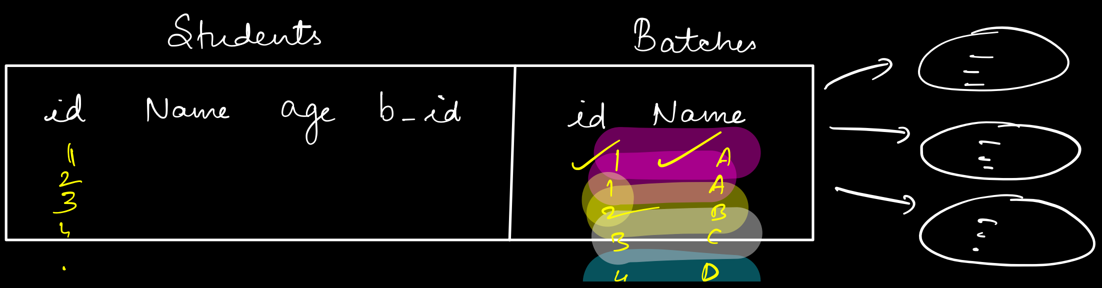
Q Print the batch name with count of students that have more than 100 students.

Students

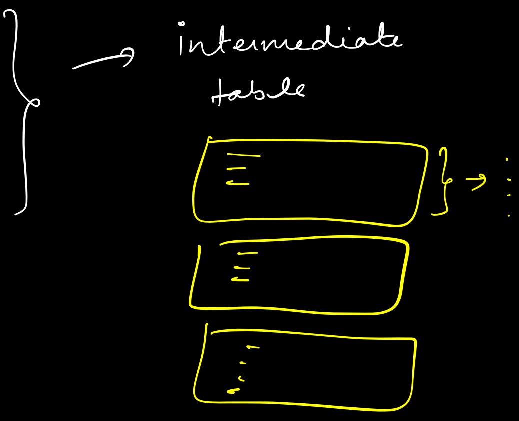
| id | Name | age | b_id |
|---|---|---|---|

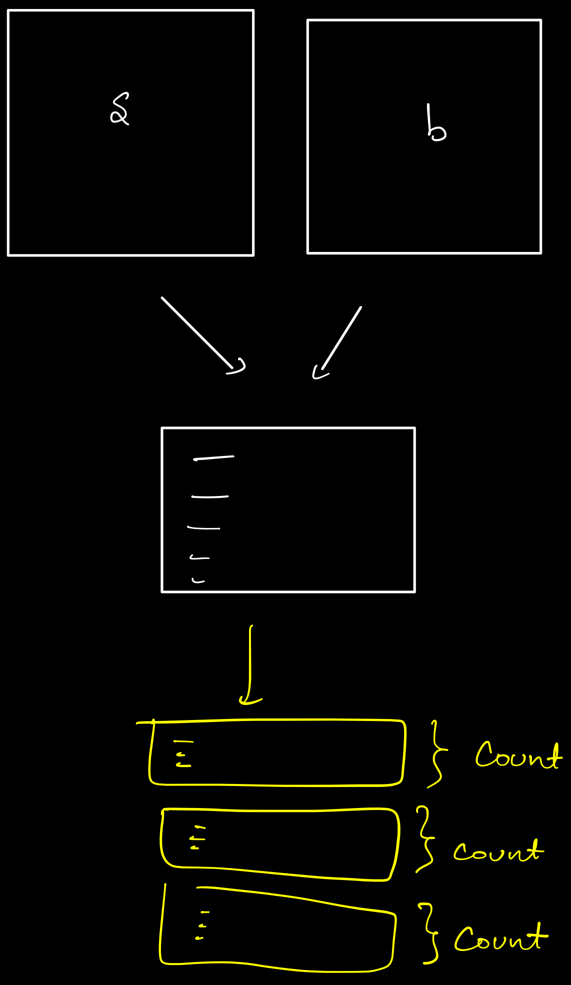
Batches

| id | Name |
|---|---|



```
Select  count(s.id), b_name
from  Students s
join  Batches b
on  s.b_id = b.id
group by  b_name
```





| b_name | Count |
|---|---|
| A | 100 |
| B | 125 |
| C | 180 |
| ~~D~~ | ~~90~~ |
| E | 110 |

→ Have we been asked to print all the groups.

WHERE → used to filter rows, not groups

HAVING :- allows you to filter groups

```
Select count(s.id), b_name
from Students s
join Batches b
on s.b_id = b.id
where clause ⇒
group by b. name
having count(s_id) > 100
```

A
B
C
E

```
A =

B =

ans1 =

for each row in A

    for each row in B

        if (con' is T)

            ans1 . add (row1 + row2)


ans2 = [ ]
for each row in ans1

    if (cond is T)

        ans2 . add (row)

        < b_name, list <rows> >
Map < Group by col > b_name_groups

for each row in ans2

    b_name_groups[row[b_name]].insert(row)
```

$\Longrightarrow$ from (join on con) $\swarrow$

WHERE (filter rows) $\swarrow$

GROUP BY $\swarrow$

```
for each group in b_name.groups
    if(cond is T){                    } HAVING
        print(b_n_g[...])
```

$$\boxed{\text{You can't use where after groups}}$$

$$\begin{tikzcd}
\text{FROM} \arrow[d] & \text{(join)} \\
\text{WHERE} \arrow[d] & \text{(rows filter)} \\
\boxed{\text{GROUP BY}} \arrow[d] & \\
\text{HAVING} \arrow[d] \arrow[r] & \text{ORDER BY} \\
\text{SELECT} &
\end{tikzcd}$$