Agenda

- Aggregates
- Built-In functions
  - Numbers
  - Strings
  - Date
- Conditions
  - IF
  - IF NULL
  - CASE
  - COALESCE

Starting At 7:08 AM

Aggregation is grouping / collecting / combining

- max()
- min()
- avg()
- sum()
- count()

Students

| id | Name | batch_id | psp |
|---|---|---|---|
| 1. | Alok | 1 | 80 |
| 2. | Sumit | 3 | 60 |
| 3. | Manju | 3 | 90 |
| 4. | Neha | 2 | 90 |
| 5. | Rahul | 1 | 80 |

Q1:- Find avg psp of students. (Easy question)

```
select avg(psp)
from students;
```

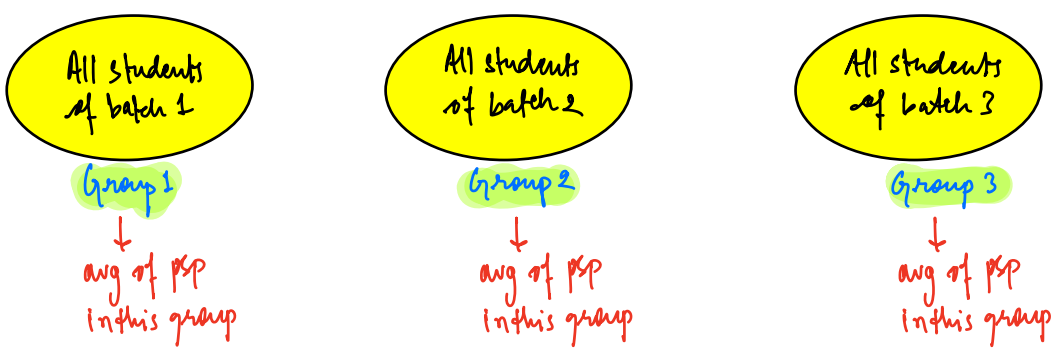
Q2:- Find avg psp of students of every batch.

O/P →

| batch_Id | avg_psp |
|---|---|
| 1 | 80 |
| 2 | 90 |
| 3 | 50 |

Steps

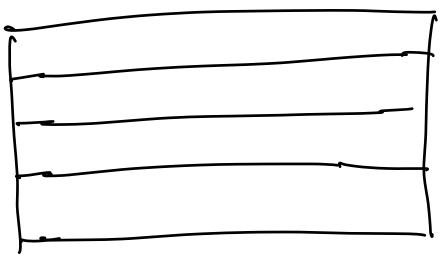
(1) Group diff. rows of the table.



## GROUP BY

stu_id

select col in group or aggregate fn. batch_id, avg(psp)
from students
group by batch_id



① Group by groups the rows by column.
② It calls the aggregate function on each group.

Students

| id | Name | batch_id | psp |
|---|---|---|---|
| 1. | Alok | 1 | 80 |
| 2. | Sumit | 3 | 60 |
| 3. | Manju | 3 | 90 |
| 4. | Neha | 2 | 90 |
| 5. | Rahul | 1 | 80 |

Q Find batches where avg. psp of all the students in that batch >= 80.

Approach 1:-

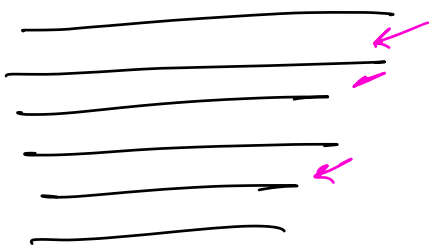
```
Select batch_id, avg(psp)
from students
group by batch_id
where avg(psp)>=80
```

✗ Won't compile.

| batch_id | avg(psp) |
|---|---|
| 1 | 80 |
| 2 | 90 |
| 3 | 50 |

Approach 2:-

```
Select batch_id, avg(psp)
from students
where psp>=80
group by batch_id
```



Where picks all rows with psp>=80

e.g.

| stu_id | psp | batch_id |
|---|---|---|
| 1 | 60 | 2 |
| 2 | 90 | 3 |
| 3 | 70 | 1 |
| 4 | 90 | 3 |
| 5 | 80 | 2 |
| 6 | 90 | 1 |

| batch_id | avg(psp) |
|---|---|
| 1 | (90+70)/2 = 80 |
| 2 | (80+60)/2 = 70 ✗ |
| 3 | (90+90)/2 = 90 |

| batch_id | avg(psp) |
|---|---|
| 1 | 90 ✗ |
| 2 | 80 |
| 3 | 90 |

| batch_id | avg(psp) |
|---|---|
| 1 | 80 |
| 3 | 90 |

## HAVING

→ Queries where we want to use WHERE clause after GROUP BY.

→ We can use HAVING clause to filter after GROUP BY.

```
Select batch_id, avg(psp)
from students
group by batch_id
having avg(psp)>=80
```

① FROM

② WHERE // picks up the rows in a table according to the conditions defined in the where clause.

③ GROUP BY // groups rows which are o/p from ② according to the condition in group by.
If no WHERE statement, pick all rows and group them.

④ HAVING // filtering on groups
// filtering can be done based on aggregate fn. OR based on cols in your group by.

Q group by OR order by ?
↓
comes first.

| WHERE | HAVING |
|---|---|
| 1) filters rows of the original table. | 1) filters groups. |
| 2) comes before group by | 2) comes after group by |
| 3) filters by cols/attributes of your table. | 3) filters by aggregate fns. or by cols. via which table was grouped. |

Q. Find avg psp of every batch that has more than 5 students.

```
select batch_id, avg(psp)
from students
group by batch_id
having count(stu_id) > 5
```

## Built-In functions.

Numbers =
Strings
Date. [Important one]

## NUMBERS

① ROUND (number, digits after decimal)
e.g. ROUND(2.31738, 2) ⇒ 2.32

② TRUNCATE (number, digits after decimal)
e.g. TRUNCATE(2.31738, 2) ⇒ 2.31

③ CIELING → neat integer >= number
e.g. CIELING(3) = 3
CIELING(2.94) = 3
CIELING(2.1) = 3

④ FLOOR → closest integer <= number
e.g. FLOOR(3) = 3
FLOOR(2.94) = 2
FLOOR(2.1) = 2

⑤ ABS → absolute values of any number.
e.g. ABS(2.94) = 2.94
ABS(-2.94) = 2.94

⑥ RAND → no parameter. Returns a random value in the range 0 to 1 (both inclusive)

Q. Build a function that generates kind of random values b/w [0 to 10] these values have to be integers.

FLOOR(RAND()*10)

0 – 0.0999 → 0
0.1 – 0.1999 → 1
0.2 – 0.299 → 2
⋮
0.9 – 0.999 → 9
1 → 10

ROUND(RAND(),1)*10

0.00.. – 0.04999 → 0
0.05 – 0.14999 → 0.1 → 1
0.15 – 0.24999 → 0.2 → 2
⋮
0.85 – 0.94999 → 0.9 → 9
0.95 – 1 → 1 → 10

## Strings.

1. LENGTH(String)
2. UPPER(String) → converts any string to UPPERCASE
   e.g. UPPER("DElhi") = DELHI
3. LOWER(String) → converts string to lowercase.
   e.g. LOWER("Delhi") = delhi
4. LTRIM(String) → Removes spaces from left side in a string
5. RTRIM(String)
6. TRIM(String) → Removes spaces from both sides in string
7. LOCATE(tofind, whereString)
   - tofind: string you want
   - whereString: string where you searching for

   e.g. find if "el" exists in "Delhi"

   LOCATE("el", "Delhi") → 2 (positions: D=1, e=2, l=3, h=4, i=5)
   - → 0, if doesn't exist
   - → if exists, it returns starting indexing.

Note:- string indexing in MySQL starts at 1

8. CHUNK
   Extracts a chunk of a string
   e.g. "DD/MM/YY" (positions 1 2 3 4 5 6 7 8)

   Q. Print date for every row.

   8.1 LEFT(String, length) → gives you length size chars from start
   e.g. LEFT(DD/MM/YY, 2) → DD

   8.2 RIGHT(String, length) → gives you length size chars from end.
   e.g. RIGHT(DD/MM/YY, 2) → YY

   8.3 SUBSTRING(String, start, length) → Generic substring function
   e.g. SUBSTRING(x, 4, 2) → MM

## DATE AND TIME

1. NOW() → prints the current date and time
2. CURDATE()
3. CURTIME()
4. YEAR(date) → gives you year part from date.
   e.g. YEAR(2022/10/22) = 2022.

5. MONTH(date) → October
6. DAYNAME(date) → Monday

→ Add something to a date
→ Sub - - -
→ find diff. b/w dates.

1. date_add(date, interval)
   ↳ interval 10 minute
   interval 1 day.

2. date_sub(date, interval)

| id | survey_form_type | survey_form_link | date_submitted |
|---|---|---|---|

Q. Write SQL query to get all the forms submitted in last 24 hours.

```
select id
from surveys
where date_submitted >= date_sub(now(), interval 1 day)
```

③ datediff (date1, date2) → diff b/w 2 dates.

e.g. datediff (28/10/2022, 30/10/2022) ⇒ 2

Q Find all orders where payment was done >= 2 days after placing the order.

```
select * from orders
where datediff(payment_at, placed_at) >= 2
```

Tuesday → Transactions and Indexing (conditions)

Wednesday