

1. Good Evening
  2. Lecture begins at 9:10pm
  3. Topic - Aggregates & Builtin Fns.
- 

## Agenda

1. Aggregates. ✓
  2. GROUP BY - ? ✓
  3. HAVING ✓
  4. Builtin - Fns.
- 

Aggregates - MAX, MIN, SUM, AVG, COUNT

$$\text{SUM}(1, 2, 5, 10) = 18$$

$$\text{COUNT}(1, 2, 5, 10) = 4$$

$$\text{MAX}(1, 2, 5, 10) = 10$$

$$\text{MIN}(1, 2, 5, 10) = 1$$

$$\text{AVG } (1, 2, 5, 10) = 4.5$$

Students

id	name	bid	psp		
1	A	1	90	✓	✓
2	B	1	80	✓	✓
3	C	1	70	✓	✓
4	D	2	60	✓	x
5	E	2	50	✓	x
6	F	3	30	✓	x

→ Give AVG psp of all students

$$\begin{aligned} & \text{SELECT AVG (psp)} \\ & \text{FROM Students} \end{aligned} \quad \frac{580}{6}$$

→ Avg psp of 1<sup>st</sup> batch students.

$$\begin{aligned} & \text{SELECT AVG (psp)} \\ & \text{FROM Students} \\ & \text{WHERE bid} = 1 \end{aligned}$$

```

SELECT  name, AVG (psp)
FROM    students

```

X  
↓

Doesn't  
work  
with a  
where  
clause also.

A	389/6
B	"
C	"
D	"
E	"
F	"

X

**RULE** → In select clause either have only column names or only aggregats. Mixing aggregats & columns is not allowed (without a groupby clause)

★ Aggregats don't work on NULL values.

COUNT (1, 2, NULL, 10, 20) = 4

$$\text{AVG} (1, 2, \text{NULL}, 10, 20) = \frac{33}{4} \\ = 8.75$$

Finding rows in a table

```
SELECT COUNT(points)
FROM customers
```

should be a non-null column

```
SELECT count(customer-id), count(*)
FROM customers;
```

count(T) → Constant

customer-id	points
✓ 1 ✓	10 ✓
✓ 2 ✓	20 ✓
✓ 3 ✓	NULL ✓
✓ 4 ✓	40 ✓

SELECT SUM (Salary)

FROM employee

WHERE name = 'Akash'

GROUP BY

Students

id	name	bid	psp	bid	avg psp
1	A	1	90	1	80
2	B	1	80		
3	C	1	70		
4	D	2	80	2	85
5	E	2	90		
6	F	3	50	3	50
7	G	3	40		
8	H	3	60		

Avg psp of all students

SELECT AVG (psp)

FROM Students

Avg psp of every batch.

```
SELECT bid, AVG (psf)
FROM students
GROUP BY bid;
```

Rule — In the select clause of a group by query, we can have the grouping column as it is. Any other column in the select clause will come with an aggregate fn.

---

BREAK = 10:08 to 10:17

## Students

id	name	bid	psp
1	A	1	60
2	B	2	70
3	C	3	80
4	D	1	90
5	E	2	50
6	F	3	60
7	G	1	70
8	H	2	80

bid	avg psp
1	220/3
2	200/3
3	70

Give Batches & their avg psp  
 batches (with at-least 3 students)

SELECT bid, AVG(psp)

FROM Students

GROUP BY bid

HAVING COUNT(id) ≥ 3;

→ Find avg psp of <sup>(above 80 psp students)</sup> all batches  
 having at-least 3 students above  
 80 psp

```
SELECT bid, AVG(bsp)
FROM Students
WHERE bsp ≥ 80
GROUP BY bid
HAVING COUNT(id) ≥ 3
```

WHERE	HAVING
→ Before group by	→ After group by
→ where filters rows.	→ Having filters groups.
→ <u>Can't use aggregates</u>	→ Uses aggregates.



## GROUP BY (Multiple columns)

Students

	id	name	bid	mid	psp
1 90, 80 + 70, 90	1	A	1	1	90
	2	B	1	1	80
	3	C	1	2	70
	4	D	1	2	90
2 80, 70	5	E	2	1	80
	6	F	2	2	70

Group 1 (bid=1): 90, 80, 70, 90  
 Group 2 (bid=2): 80, 70  
 Group 1 (mid=1): 90, 80  
 Group 2 (mid=2): 70, 90, 70

<pre>SELECT bid, A(psp) FROM students GROUP BY <u>bid</u></pre>	<pre>SELECT mid, A(psp) FROM students GROUP BY mid</pre>
---	--

```
SELECT bid, mid, AVG(psp)
FROM students
GROUP BY (bid, mid);
```

1	1	90 + 80
1	2	70 + 50
2	1	80
2	2	70

---

## Compound Joins

Tbl 1				Tbl 2		
id	c1	c2		id	c1'	c2'
1	1	2		1	2	1
2	2	3		2	1	2
3	3	1		3	5	5
4	1	2		4	1	3
5	2	4		5	4	1

SELECT \*

FROM Tbl 1 t1

JOIN Tbl 2 t2

ON  $\left[ \begin{array}{l} t1.c1 = t2.c1' \\ t1.c2 = t2.c2' \end{array} \right] \text{ AND}$

4	1	2	2	1	2
<hr/>					
1	1	2	2	1	2