Agenda.

1) Select Query functions. (Continued) ✓

2) Joins

3) Aggregates.

Announcement

20th October. → Holiday

↳ 19th Oct. → Wed. } Class.

18th Oct → Tues.

Students table

fName, lName, batchId, psp

Get all students of batch_id = 2 where psp >= 60 and psp <= 90

Send your answers.

Between → when you have statements like a >= x AND a <= y

```
select * from students where batch_id = 2 AND
(psp Between 60 and 90)
```

Note:- Between is >= and <= . inclusive.

LIKE Operator

Whenever we compare strings we have diverse usecases.

e.g. Scaler Batches table

Let's say you have only this table.

→ Get all the batches that started in August.

Batches.

id	Name
1	August Intermediate 2022
2	Aug Adv 22
3	May - -
4	June - -
5	July - -

Solution:-

1) % (percent)
2) _ (underscore)

```
select * from batches
where name LIKE 'hello';
```

→ Gets me all the rows with name = 'hello'
It does exact match.

e.g. 1) select * from batches
where name LIKE '%hello%';

% means any # of characters.
↓
>= 0

hi hello Alok

hi hello hello Alok

hi hello Alok

hello

2) _ means any single char.

name like '_ hello %'

hello I am → No.

hi hello Alok → No.

I hello Alok → Yes

↓
one char.

* The most common usage of LIKE

⇒ Find all containing 'x' in all the data.

→ Select * from table where column LIKE '%x%'

IS NULL.

IS NULL → If you have to check the value of a column to be null.
Don't say x = NULL. Rather use IS NULL

ORDER BY

By default, data sorted in ascending according to PK.

Query: Get all students where batch-id = 2 and sort by psp in descending order

Note:- You can orderBy style/ multiple columns.

```
select *
from students
orderBy (psp, timeOfSolving, fName)
```

	psp	timeof solving (mins)
Karthick	90	30
Srikanth	90	30

* By default, orderBy sorts in ascending order.
* If a tie happens after all columns mentioned in order by clause, the tie breaker happens with PK.
* To sort in descending order.

```
select *
from students
order By (psp) Desc
```

LIMIT

```
select * from students
Limit 5;
```
→ first 5 students based on PK

```
select * from students
orderby psp
limit 5
```
→ bottom 5 students by psp

```
select * from students
order by psp desc
limit5
```
→ top 5 students by psp

JOINS

Scaler DB.

Students

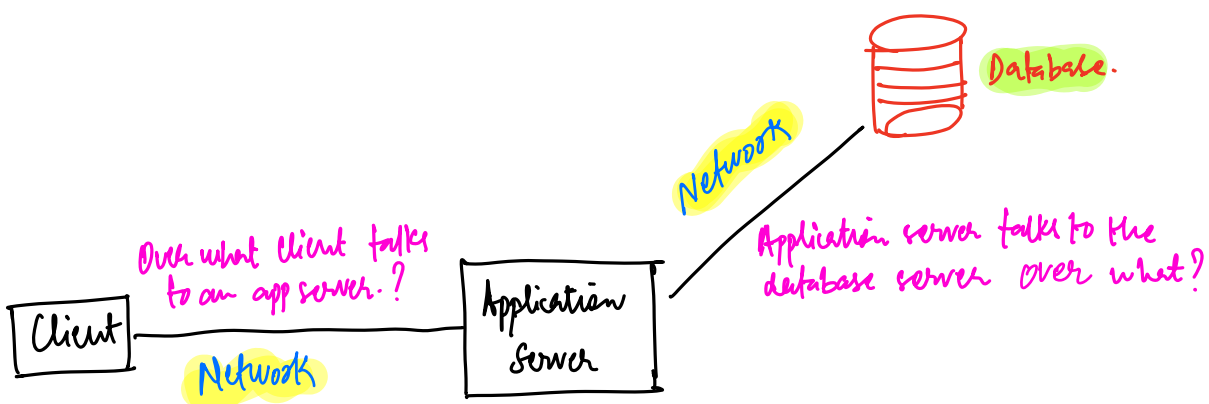
id	Name	email	password	batch_id

Batches

id	Name

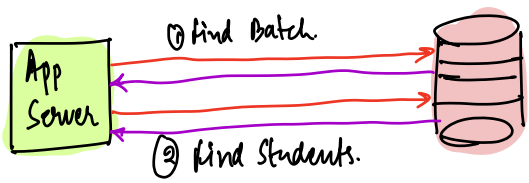
Ques:- Find all the students in 'Apr21 Intermediate' Batch.

Solution:- a) Go to batches table and find the batch_id with given name.
b) Find all the students with the batch_id. in students table.



Each N/W call → 250ms.

total time → 1s.



Stu id.	st. Name	st. Email	st Batch Id	st. Batch Name.

Bad Schema Table (Denormalized)

Issue:- Redundancy.

Benefit:- Queries will be faster.

Break:- 8:40.

We have these 2 tables.

Students

id	Name	email	password	batch_id
1.	Alok	alokgmail	123	1
2.	Sumit	sumit	456	1
3.	Manju	manju	8910	1
4.	Neha	neha	Abc	2
5.	Rahul	rahul	naam SunaHoga	1

Batches

id	Name
1	Dec 21
2	Feb 22

Students JOIN BATCHES.

Students

id	Name	batch_id
1.	Alok	1
2.	Sumit	1
3.	Manju	1
4.	Neha	2
5.	Rahul	1

Batches

id	Name
1	Dec 21
2	Feb 22

JOIN

id	Name	St_batch_id	batch_id	batch_name
1.	Alok	1	1	Dec 21
2.	Sumit	1	1	Dec 21
3.	Manju	1	1	Dec 21
4.	Neha	2	2	Feb 22
5.	Rahul	1	1	Dec 21

↳ # col = # of cols in Students + # cols in Batches.

Select * from 'this joined table'
where batch_name = 'Dec 21'

Types of Joins.

i) Inner Join → combines rows of both tables based on given condition.

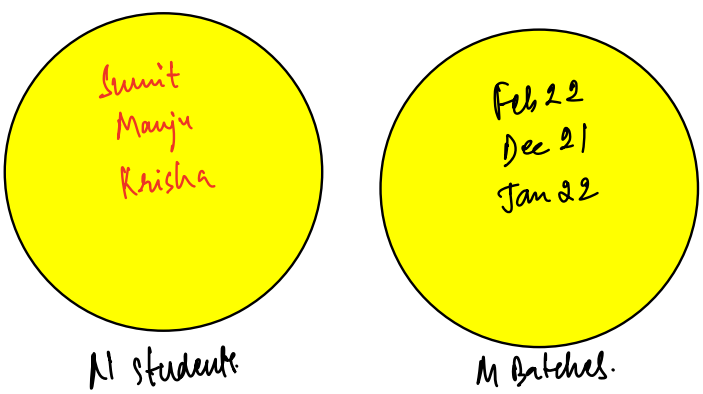
Students.

id	Name	batch_id

Batches

id	Name

```
Select *
from students
inner join batches
on students.batch_id = batches.id;
where batches.name = "Dec 21"
```



for every row of students:
  for every row of batches:
    if the join condition is satisfied:
      pick and put this merged row in o/p.

T.C = O(N*M)

Let's take an example.

Students table

id	name

Phone numbers table

st_id	phone_num

Q. Print all the students with all their phone nos.

```
select name, phone_num
from students
inner join phone_numbers
on students.id = phone_numbers.std_id;
```

join

2) Self Join

Students.

id	name	peer-reviewer_id
1	Manju	4
2	Krishna	4
3	Rahul	1
4	Rekha	2

PeerReviewers Table.

id	Name

→ X

Q. Print the name of every student with the name of their peer.

O/p format →

id	name	peer_name

this is how I want the o/p.

```
select *
from students s
join peerReviewer p
on s.peerId = p.id
```

→

```
Select id, s.name, p.name
from students s
join students p
on s.peerId = p.id
```

Self join is nothing but inner join with same table.

Students.

id	name	peer-reviewer_id
1	Manju	4
2	Krishna	4
3	Rahul	1
4	Rekha	2

Students.

id	name	peer-reviewer_id
1	Manju	4
2	Krishna	4
3	Rahul	1
4	Rekha	2

Do a inner join b/w these 2 tables (Same table) → self join.

3) Compound Joins

Joining multiple tables.

Scaler DB

Students

id	Name	batch_id

batches

id	ins_id

instructors.

id	Name

Ques Print st_Name, ins_Name for every student

Ques

Students

id	Name	batch_id
1.	Alok	1
2.	Sumit	1
3.	Manju	NULL
4.	Neha	2
5.	Rahul	1

Recently joined. No batch.

Batches

id	Name
1	Dec 21
2	Feb 22
3	Jan 22

Print the name of every student with the name of the batch.
If a student is not present in any batch, print their name with NULL as the batch name.

Outer Joins

- Left Joins
- Right Joins
- Full Joins.

1) **Left Join** → If a row of left side doesn't match with any row of right side, it will be added in final answer with NULL.

Students

id	Name	batch_id
1.	Alok	1
2.	Sumit	1
3.	Manju	NULL
4.	Neha	2
5.	Rahul	1

Recently joined. No batch.

Batches

id	Name
1	Dec 21
2	Feb 22
3	Jan 22

↓ Left Join these 2 tables.

id	Name	St_batch_id	batch_id	batch_name
1.	Alok	1	1	Dec 21
2.	Sumit	1	1	Dec 21
3.	Manju	NULL	NULL	NULL
4.	Neha	2	2	Feb 22
5.	Rahul	1	1	Dec 21

2) Right Join → All rows of right side are returned even if they do not match with any row of left side

Students

id	Name	batch_id
1.	Alok	1
2.	Sumit	1
3.	Manju	NULL
4.	Neha	2
5.	Rahul	1

Recently joined. No batch.

Batches

id	Name
1	Dec 21
2	Feb 22
3	Jan 22

Right Join b/w these 2 tables

id	Name	St_batch_id	batch_id	batch_name
1.	Alok	1	1	Dec 21
2.	Sumit	1	1	Dec 21
4.	Neha	2	2	Feb 22
5.	Rahul	1	1	Dec 21
NULL	NULL	NULL	3	Jan 22

3) FULL JOIN

If any row of any side is not present put that with a NULL.

Thursday

→ Complete Joins

→ Start Aggregates and Built-in functions.