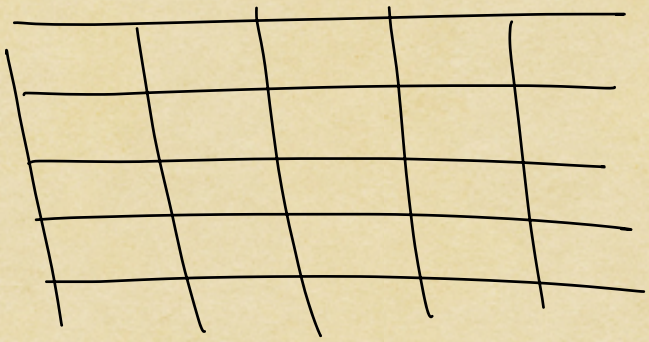
Joins

⇒ select * from table
where {=}



⇒ Is fetching data from a single table always good enough?

⇒ Scaler's DB :-

Students

id	name	email	phone no.	batch Id

Batches

id	name

* Find all the Students from batch "Apr 22 Beginner".

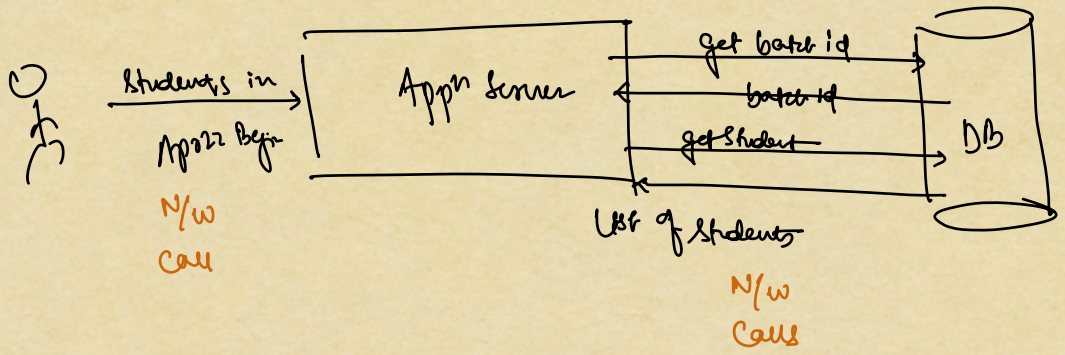
↓

batch name

Soln

a) Go to the batches table, and fetch the batch-id for the given batch name

b) Use this batch-id and fetch all the students from Students table



* N/w calls are much much slower than CPU execution.

⇒ only the n/w latency added 300 ms to the overall execution time

Students

Stid	name	phone	email	batchId	batchName

batches

but this table will reduce the n/w calls. and speed up execution

this table will cause redundancy

this table is not normalized

denormalized

* Can we actually create something like the above table, without actually denormalizing the tables (Students & batches)

Ans ⇒ Yes

↓

by Joins

Students

id	name	email	phone no.	batchId
1	Sandeep	-	1234	1
2	Hemanshu	-	5678	1
3	Rakesh	-	1624	2
4	Praveen	-	5829	1
5	Akhay	-	0012	2

Batches

id	name
1	Mar22 Begin
2	Apr22 Begin

O/P ⇒ Rakesh

Akhay

⇒ Joining | merging 2 tables into 1 w/o actually changing them physically

* join only exists for that query

* Query will feel that the tables are actually

joined together

STUDENTS JOIN BATCHES — bringing all the rows & columns parts of a specific covers together temporarily

Students

Id	name	email	phone no.	batchId
1	Sandeep	-	1234	1
2	Venkatesh	-	5678	1
3	Rakesh	-	1624	2
4	Praveen	-	5829	1
5	Akhay	-	0012	2

Batches

id	name
1	Mar22 Begin
2	Apr22 Begin

JOIN

select * from

Id	name	email	phone no.	Stu. batchId (Students)	batchId (Batches)	name
1	Sandeep	-	1234	1	1	Mar22 Begin
2	Venkatesh	-	5678	1	1	Mar22 Begin
3	Rakesh	-	1624	2	2	Apr22 Begin
4	Praveen	-	5829	1	1	Mar22 Begin
5	Akhay	-	001	2	2	Apr22 Begin

where batchName = 'b' ;

Joined table will have equal no. of columns as the no. of columns combined from tables which are joined

total columns (joined) = no. of columns (A) + no. of columns (B)

↓

A + B

Algo

for every row in Student (N rows)
    for every row in Batches (M rows)
        if the join condition is satisfied
            put a merged row in ans.

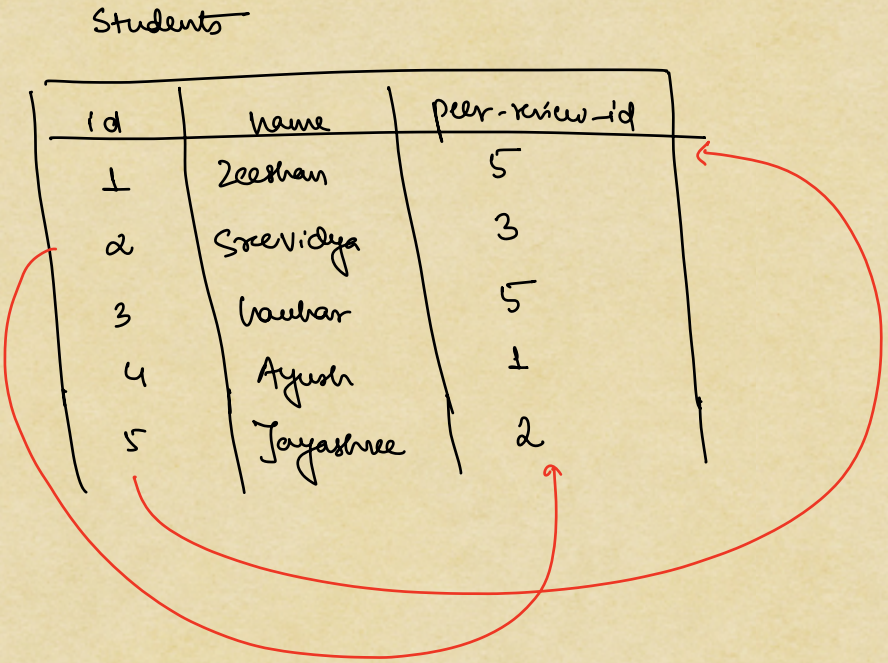
O(N*M)

* if multiple rows matches from one table to a single row in the other table, there will be multiple entries for that row in the joined table

* inner join is same as join, inner is optional.

⇒ SELF JOIN

Students

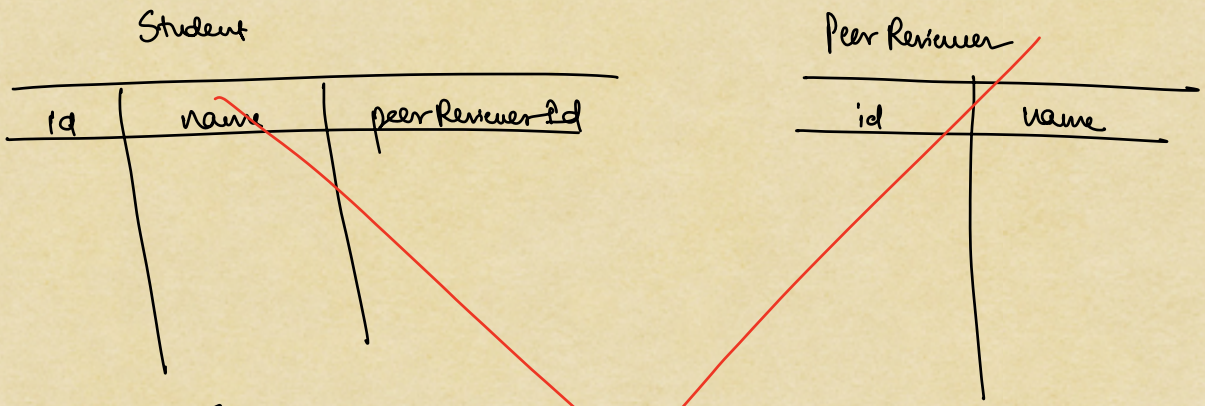


id	name	peer-review-id
1	Zeeshan	5
2	SreeVidya	3
3	Gautam	5
4	Ayush	1
5	Jayashree	2

Qn Print the name of every student with their peer-reviewers name

O/P :

StudentName	ReviewerName
Zeeshan	Jayashree
SreeVidya	Gautam
Gautam	Jayashree
Ayush	Zeeshan
Jayashree	SreeVidya




```
Select *
from Student S
join PeerReviewer P
on S.PeerRevId = P.id;
```

we dont have 2 tables

```
Select *
from Student S
join Student P
on S.PeerRevId = P.id;
```

=> Can we combine multiple conditions while joins

Ans: Yes

compound joins

```
Select *
from table1 t1
join table2 t2
on cond1 or cond2 and cond3.
```

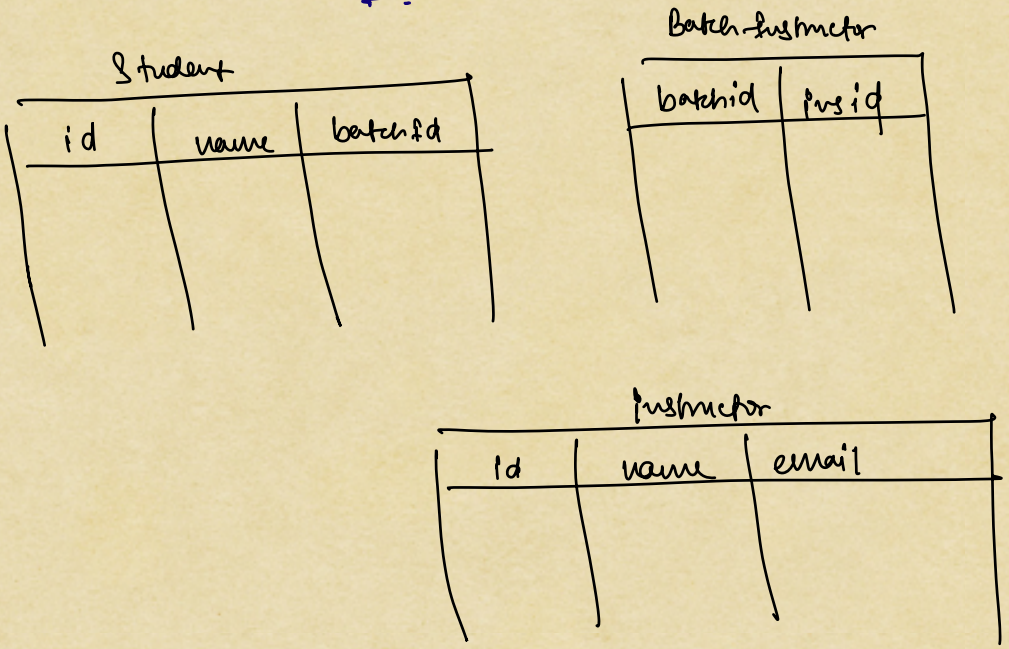
not can also be used

HW
try this out

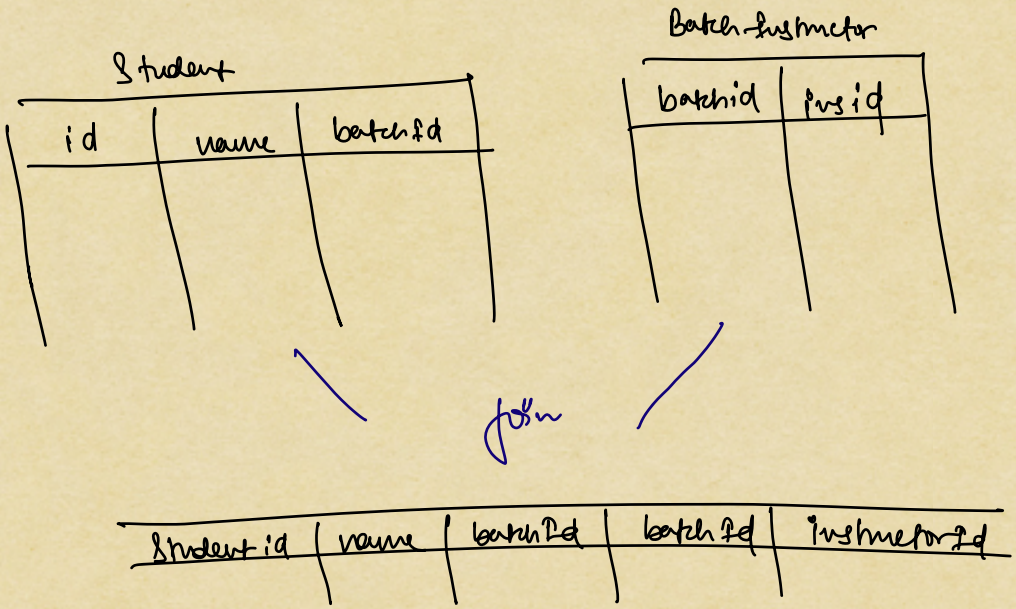
↓ and/or ↓ and/or

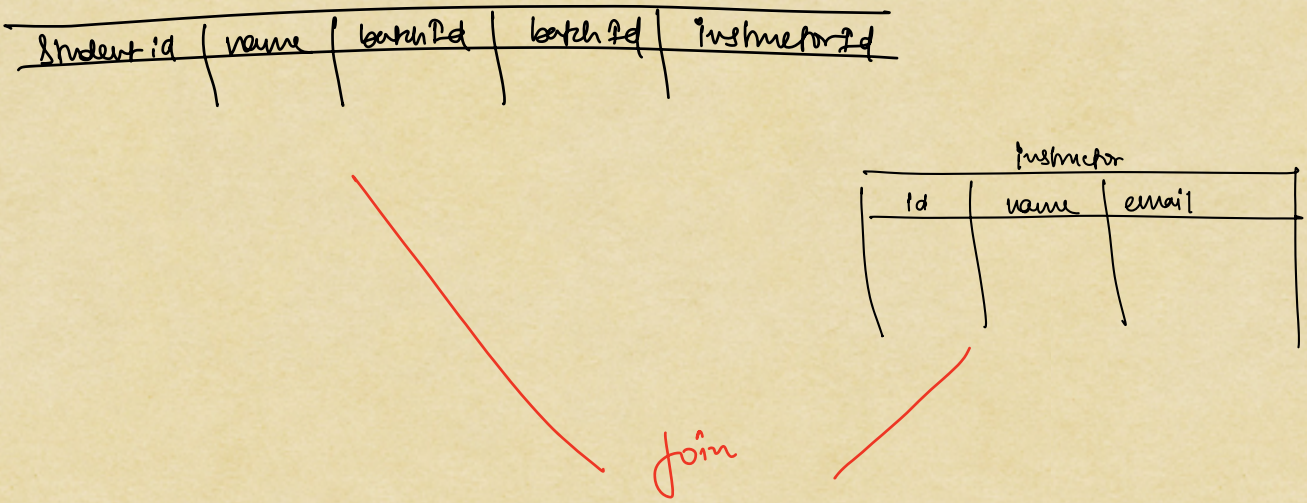
⇒ Can we join more than 2 tables ?

Ans = Yes



⇒ All students names with their respective instructor names





SQL

```
Select   S.name, i.name
from     Students S
join     Batches b
on       S.batchId = b.Id
join     Instructors i
on       b.instructorId = i.Id ;
```

Create tables & execute

N Student
M batches
O instructor

→ N^3

$O(NMO)$

very high