1. Good Evening

2. Lecture begins at 9:~~to~~05 pm

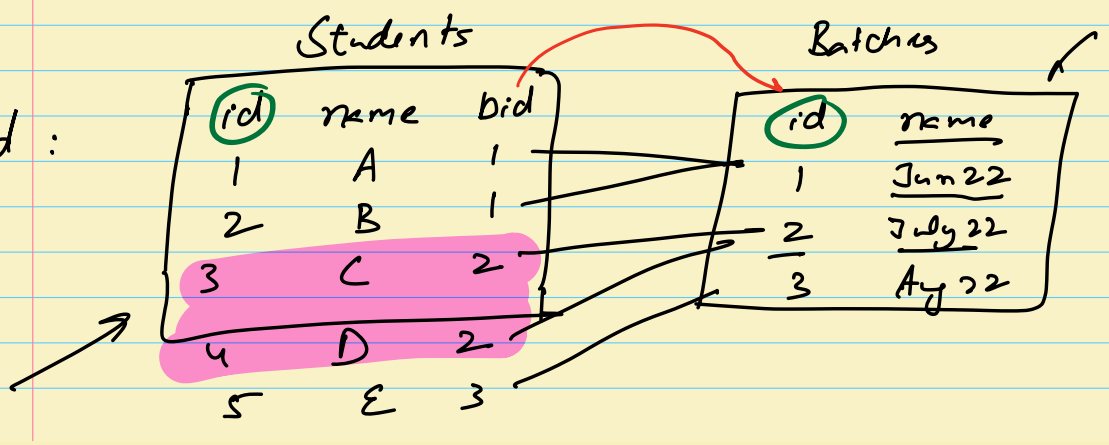
3. Topic - JOINS

Agenda : JOINS

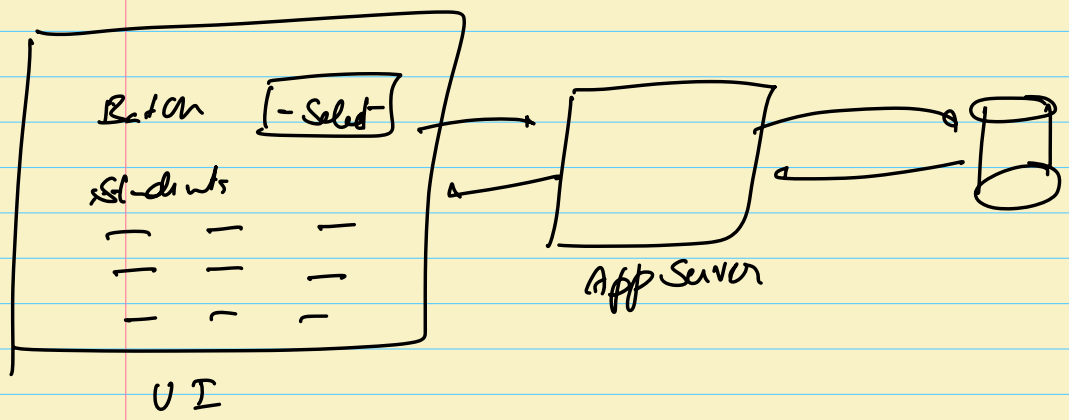
1. Inner — Intro
   - Self
   - Multi-table Joins
   - Compound
   - → Using
   - → Natural Join
2. LEFT
3. RIGHT
4. CROSS JOIN
5. Advice

Intro

Need :

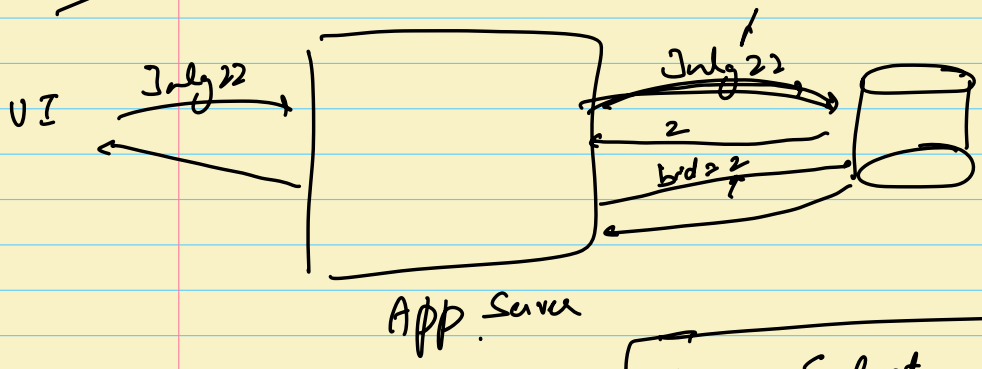


Task: Give all students belonging to "July 22" batch




```
Select *
FROM students
WHERE ——————
```

Sln 1



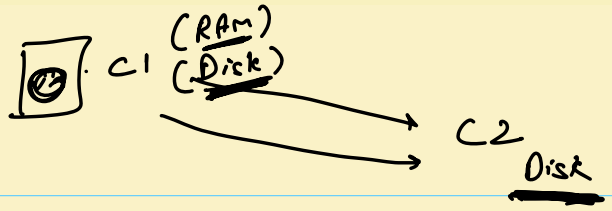
1. Select bid
from Batches
WHERE name = "July 22"

2. Select *
FROM Students
WHERE bid = 2

RAM

Disk

Network



RAM >> Disk >> Network

Cons of S1 approach → 2 calls, network being slower, this approach will be slower.

0 ✓

A1 ✓

└ Cons ✓

Sln 2

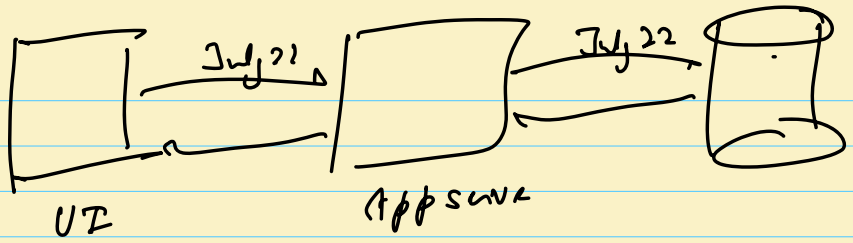
~~June 22~~

June 22

Students

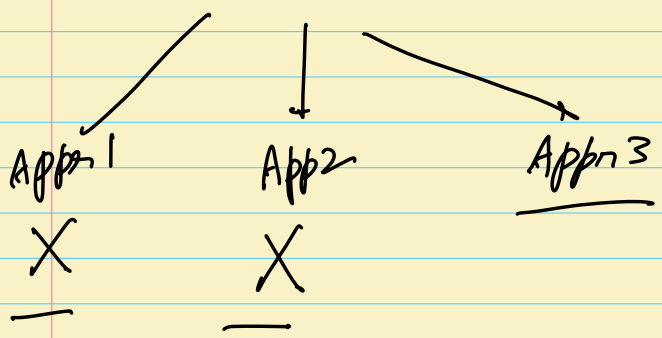
id (c)	name	bname
1	A	Jun 22
2	B	Jun 22
3	C	July 22
4	D	July 22
5	E	Aug 22

1. Space
2. Performance is low
3. Probability of data inconsistency.




```
Select *
FROM Students
WHERE bname = 'July 21'
```

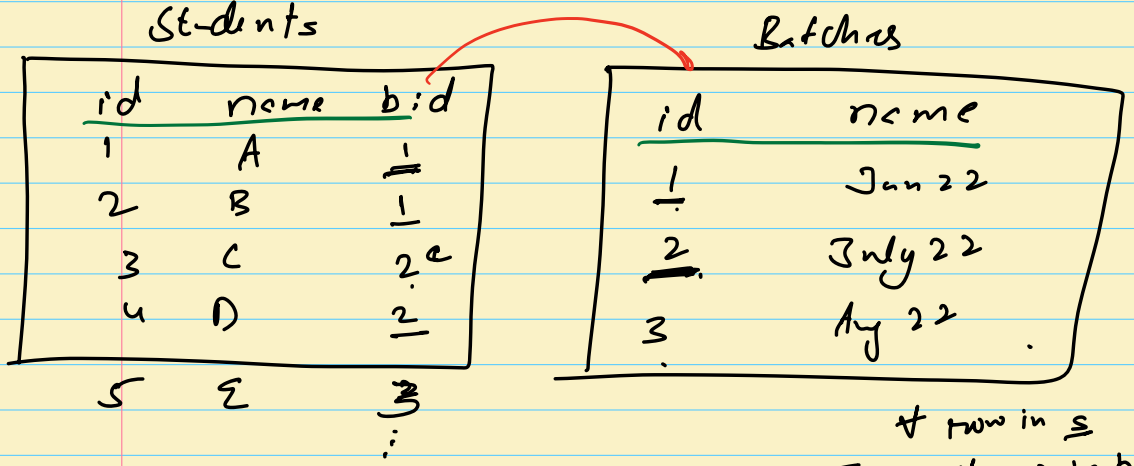
Need : Find students by batchname



Sln3 → 1. It should not have multiple calls to db

2. It should not have data with duplicacy.

JOINS



row in s × # row in b

```
SELECT  s.name, b.name
FROM    students  s
JOIN    Batches   b
    ON  s.bid = b.id
Where b.name = 'July 22'
```

SC: m × n
TC: m × n

s.id	s.name	s.bid	b.id	b.name
1	A	1	1	Jan 22
2	B	1	1	Jan 22
3	C	2	2	July 22
4	D	2	2	July 22
5	E	3	3	Aug 22

A virtual table.

C	July 22
D	July 22

```
Select  *
FROM    Students  s
JOIN    Batches   b
  ON    s.bid = b.id
```

→ SAS

eg2

Students

id	name
1	A
2	B
3	C

Phones

id	ph	sid
1	xxx	1
2	yyy	2
3	zzz	3
4	aaa	1
5	bbb	2

```
Select  *
FROM    Students  s
JOIN    Phones    p
  ON    s.id = p.sid
```

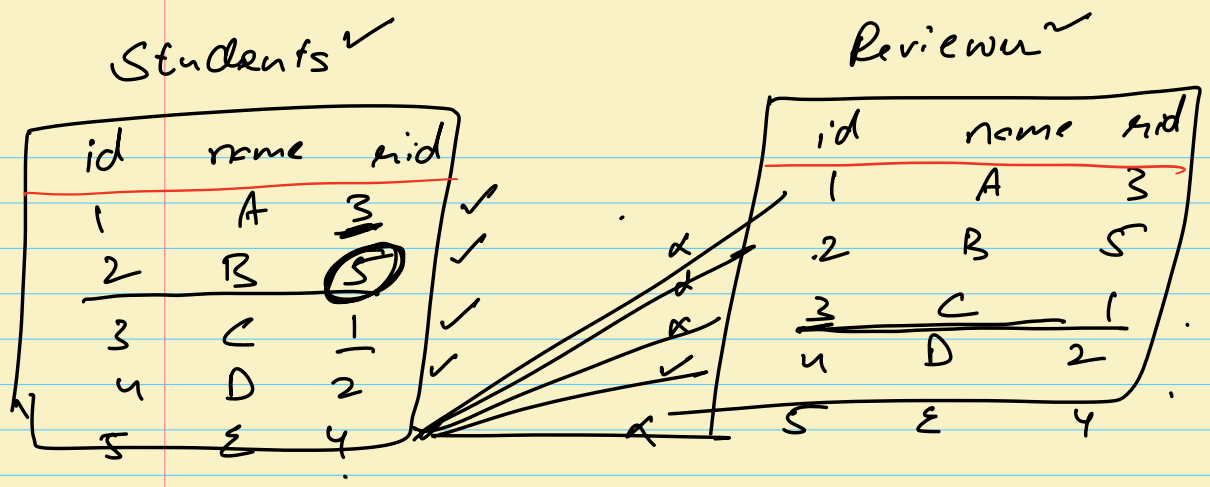
Break = 10:03 – 10:10

SELF JOIN

Students

id	name	rid
1	A	3
2	B	5
3	C	1
4	D	2
5	E	4

name	mname
A	C
B	E
C	A
D	B
E	D




```
SELECT  s.name, r.name
FROM    Students s
JOIN    Students r
  ON    s.rid = r.id
```

s.id	s.name	s.rid	r.id	r.name	r.rid
1	A	3	3	C	1
2	B	5	5	E	4
3	C	1	1	A	3
4	D	2	2	B	5
5	E	4	4	D	2

A	C
B	C
C	A
D	B
E	D

Multi-Table Joins

Students:

id	name	bid
1	A	1
2	B	2
3	C	3
4	D	1
5	E	2

Batches

id	name	iid
1	B1	3
2	B2	1
3	B3	2

Instructor

id	name
1	I1
2	I2
3	I3

s.name	i.name
A	I3
B	I1
C	I2
D	I3
E	I1

Student + Batch.

```
Select *
FROM Student s
JOIN Batches b
    ON s.bid = b.id
```

Students

s.id	s.name	s.bid
1	A	1
2	B	2
3	C	3
4	D	1
5	E	2

Batches

b.id	b.n	b.iid
1	B1	3
2	B2	1
3	B3	2
1	B1	3
2	B2	1

Instructor

id	name
1	I1
2	I2
3	I3

```
Select  s.name, i.name
FROM    Students  s
JOIN    Batches  b
 ON     s.bid = b.id
JOIN    Instructor  i
    ON  b.iid = i.id
```

Syntactical Sugar.

v1 ⟶ v2

1. Using keyword.

Students

sid	name	bid
1	A	1
2	B	2
3	C	3
4	D	1
5	E	2

Batches

bid	name
1	B1
2	B2
3	B3

Query so far

```
Select *
FROM Students s
JOIN Batches b
  ON s.bid = b.bid
```

Using keyword

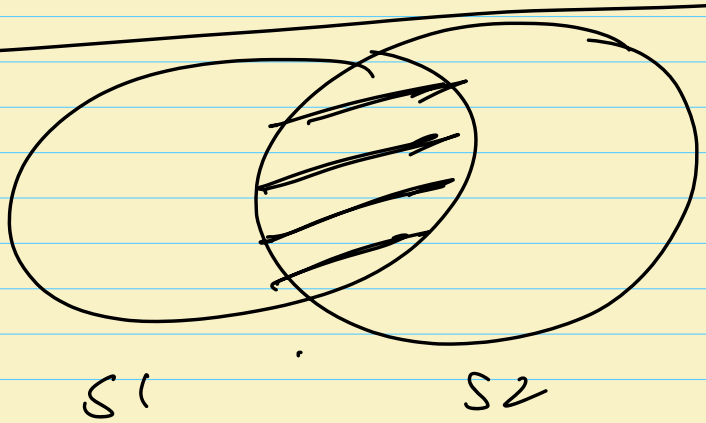
```
Select *
FROM Students
JOIN Batches
USING (bid);
```

→ when name of column is same on both sides.

2. NATURAL JOIN

```
Select *
FROM Students        → FK
NATURAL JOIN Batches → PK
```

JOIN = Inner JOIN



Batches

id	name
1	B1
2	B2
3	B3

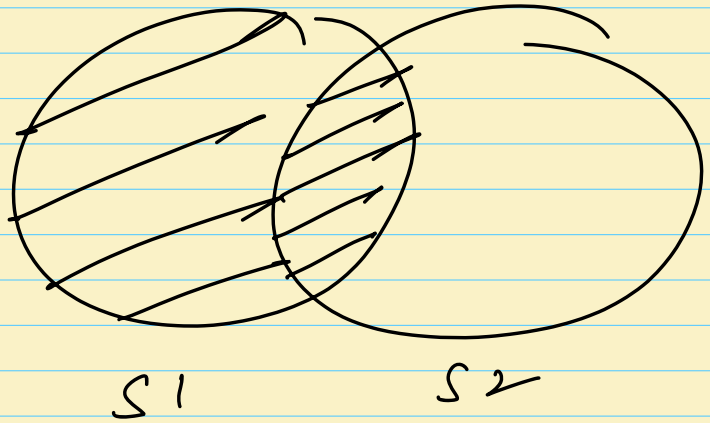
Students

id	name	bid
1	A	1
2	B	1
3	C	2
4	D	2
5	E	NULL

```
Select *
FROM Batches b
Inner JOIN Students s
    ON b.id = s.bid
```

b.id	b.n	s.id	s.n	s.bid
1	B1	1	A	1
1	B1	2	B	1
2	B2	3	C	2
2	B2	4	D	2

LEFT OUTER JOIN



```
Select *
FROM Batches b
LEFT JOIN Student s
    ON b.id = s.bid
```

b.id	b.n	s.id	s.n	s.bid
1	B1	1	A	1
1	B1	2	B	1
2	B2	3	C	2
2	B2	4	D	2
3	B3	n	n	n

1. Make result acc. to inner join
2. Add rows from LHS (if they were skipped in 1st step)

RIGHT JOIN → 1. Get inner join result
2. Add missing rows of RHS to result.

```
Select *
FROM Batches b
RIGHT JOIN Student s
    ON b.id = s.bid
```

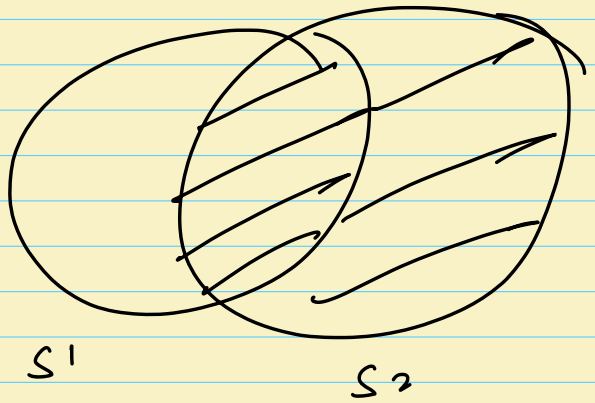
Batches

id	name
1	B1
2	B2
3	B3

Students

id	name	bid
1	A	1
2	B	1
3	C	2
4	D	2
5	E	null

biid	b.n	s.id	s.n	s.bid
1	B1	1	A	1
1	B1	2	B	1
2	B2	3	C	2
2	B2	4	D	2
null	null	5	E	null



Join = Inner JOIN

LEFT JOIN = Left outer JOIN

(Inner join + skipped rows of LHS)

RIGHT JOIN = Right Outer Join

(Inner join + skipped row on RHS)

Batches

id	name	iid
1	B1	1
2	B2	2
3	B3	null

Instructor

id	name
1	I1
2	I2
3	I3

```
Select *
FROM Batches b
LEFT JOIN Instructor i
   ON b.iid = i.id
```

b.id	b.n	b.iid	i.id	in
1	B1	1	1	I1
2	B2	2	2	I2
3	B3	null	n	n

→ FULL OUTER JOIN [Missing in MySQL]

1. Inner Join = ?

[2. Missing Rows of LHS = ?

3. " " " RHS = ?]

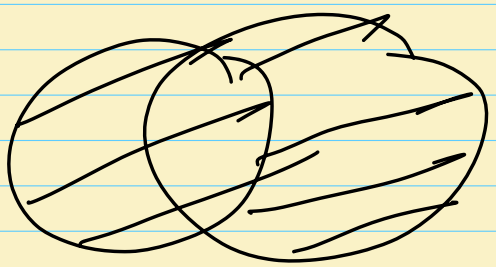
Batches

id	name
1	B1
2	B2
3	B3

Students

id	name	b.id	
1	A	1	? ✓
2	B	1	? ✓
3	C	2	? ✓
4	D	2	? ✓
5	E	n	? ✗

b.id	b.n	s.id	s.n	s.bid	
1	B1	1	A	1	FJ
1	B1	2	B	1	
2	B2	3	C	2	
2	B2	4	D	2	
3	B3	n	n	n	LJ
n	n	5	E	n	RJ



MySQL

```
SELECT *
FROM Batches b
LEFT JOIN student s
    ON b.id = s.bid
UNION

SELECT *
FROM Batches b
RIGHT JOIN student s
    ON b.id = s.bid
```

CROSS JOIN

↳ cartesian product

Students

id	name	bid
1	A	1
2	B	1
3	C	2

3 × Batches

id	name
1	B1
2	B2

2 = 6

s.id	s.n	s.bid	b.id	b.n
1	A	1	1	B1
1	A	1	2	B2
2	B	1	1	B1
2	B	1	2	B2
3	C	2	1	B1
3	C	2	2	B2

```
Select *
FROM Students
CROSS JOIN Batches
```

```
Select *
FROM Students, Batches.
```

Advice.

```
Select *
FROM Students
JOIN Batches b
  ON s.bid = b.id
WHERE b.name = "July 22"
```

```
Sele *
FROM Students s, Batches b
WHERE s.bid = b.id
  AND b.name = "July 22"
```

Batches

id	name
1	June 22
2	July 22
3	Aug 22

Students

id	name	bid
1	A	1
2	B	1
3	C	2
4	D	2
5	E	null

1	June 22	1	A	1	✓
1	June 22	2	B	1	✓
2	July 22	3	C	2	✓
2	July 22	4	D	2	✓

4 → 2

15 → 2