

1. Good Evening
2. Let us start at 9:07pm

GROUP BY
Subqueries

Agenda

1. Joins → Need?
 - └ Inner
 - └ Left
 - └ Right } Full Outer
 - └ Cross, **Natural**
 - └ An advice

Need → Getting information from multiple tables.

id	name	bid		id	name
1	A	1		1	B1
2	B	2	└─┬─┘	2	<u>B2</u>
3	C	1		3	B3
4	D	2	└─┬─┘		

Students Batch

★ Get information of all students belonging to batch named B2.

Soln1 → 1. Get Bid using batch name.

DECLARE BID INT;

SELECT BID = ~~BID~~ id

FROM Batches

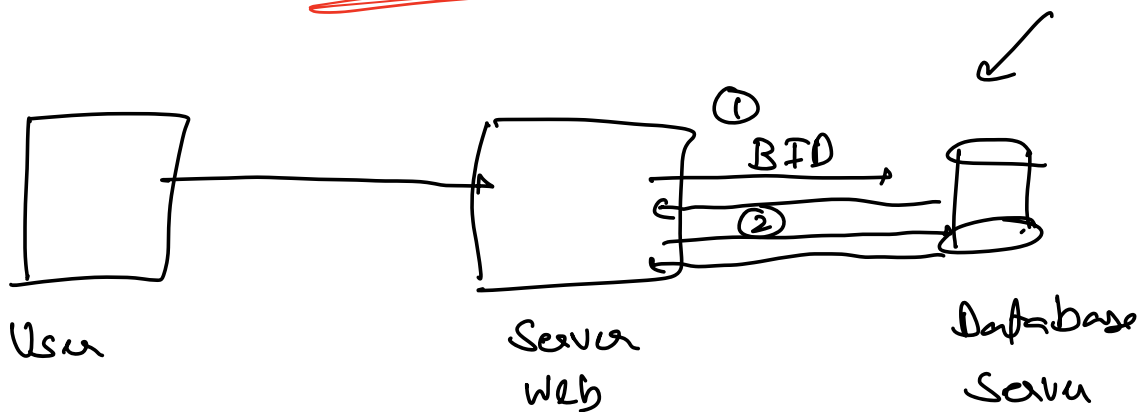
WHERE name = 'B2'

2. Get all students belonging to relevant bid

SELECT *

FROM Students

where bid = BID;



CPU + RAM >> Hard Disk >> Network

Soln 2

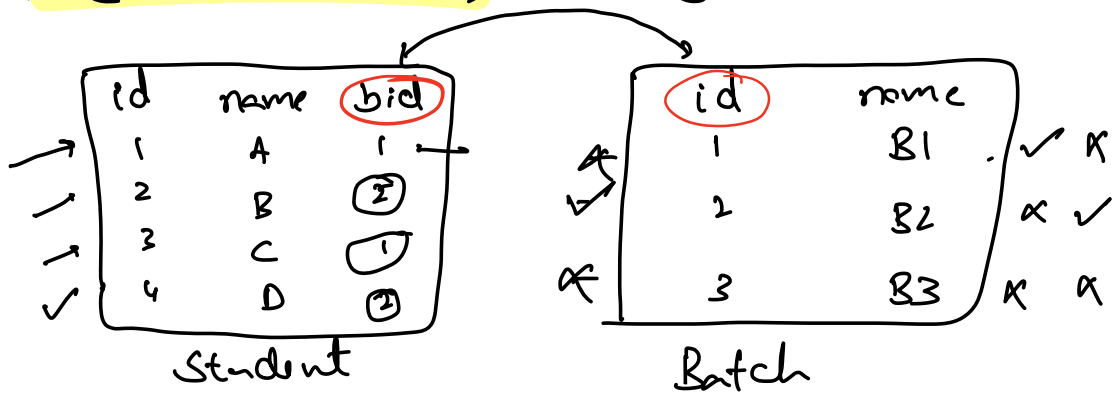
→ Very bad solution (Duplication)

id	name	bname
1	A	Aug 2022 Inter
2	B	Aug 2022 Inter
3	C	Aug 2022 Inter
4	D	Aug 2022 Inter

Students

Performance
Higher probability of inconsistent data

Soln3 (Ideal Solution) → Join



st.id	st.name	st.bid	b.id	b.name
1	A	1	1	B1
2	B	2	2	B2
3	C	1	1	B1
4	D	2	2	B2

A table "created" by join

Syntax :

```

SELECT [Students.name, Batches.name]
FROM Students
INNER JOIN Batches
ON Students.bid = Batches.id

```

id	name
1	A
2	B
3	C

Students

sid	phone
1	XYZ
1	DEF
2	ABC
3	GHI
3	JKL
3	MNO

Dinner

name	phone
A	X Y Z
A	D E F
...	

id	c1	c2	t2id
1	A	B	1
2	C	D	2
3	E	F	2
4	G	H	1

Tbl1

id	c1	c2
1	I	J
2	K	L
3	M	N

Tbl2

tbl1.id	tbl1.c1	tbl1.c2	tbl1.t2id	tbl2.id	tbl2.c1	tbl2.c2
1	A	B	1	1	I	J
2	C	D	2	2	K	L
3	E	F	2	2	K	L
4	G	H	1	1	I	J

for every row in tbl1
 {
 check every row in tbl2
 if matches make a merged row
 }

SELF JOIN : Inner join of a table with itself

E.g.

id	name	r_id
1	A	3
2	B	1
3	C	1
4	D	2

Students

Get name, reviewer-name for all students

id	name	r_id
1	A	3
2	B	1
3	C	1
4	D	2

Students

id	rname
1	A
2	B
3	C
4	D

Reviewers

```
SELECT s.name, r.rname
FROM Students s
JOIN Reviewers r
  ON s.r_id = r.id
```

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

Break till 10:11 pm

id	name	tbody
1	A	2
2	B	3
3	C	1
4	D	1

Students

name	aname
A	B
B	C
C	A
D	A

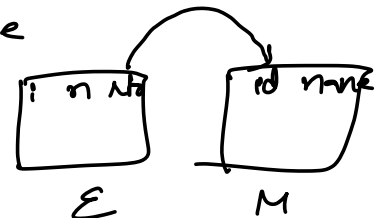
id	name	tbody
1	A	2
2	B	3
3	C	1
4	D	1

id	name
1	A
2	B
3	C
4	D

Students

Reviewer

SELECT s.name, r.name
FROM Students s
JOIN Students r
s.rby = r.id



Multi-table Joins

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

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

id	name
1	I1
2	I2

Students Batches

s name	i name
A	I1
B	I1
C	I2
D	I2

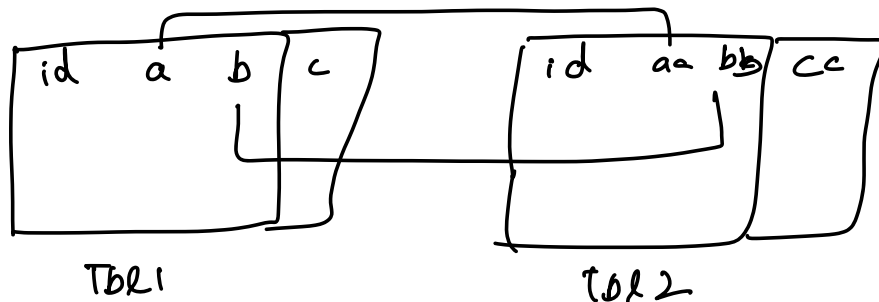
Instructor

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

```

SELECT
FROM Students s
JOIN Batches b
    ON s.bid = b.id
JOIN Instructors i
    ON b.iid = i.id
  
```

Compound Join → Multiple conditions in the on clause.



```

SELECT *
FROM Tbl1 T1
JOIN Tbl2 T2
    ON T1.b = T2.bb AND
  
```

$$T1.b = T2.bb$$

→ Using keyword
 → Natural Join

} → Syntactical
 sugars on
 inner joins.

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

Students

bid	bname
1	B1
2	B2

Batches

```

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

USING KEYWORD

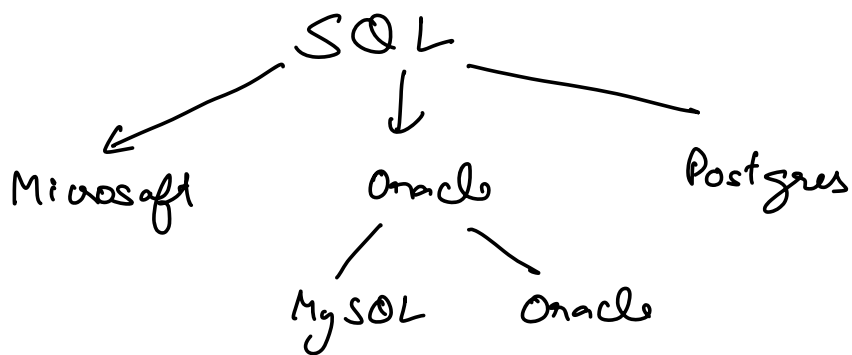
```

SELECT *
FROM Students
JOIN Batches
USING (bid)
  
```

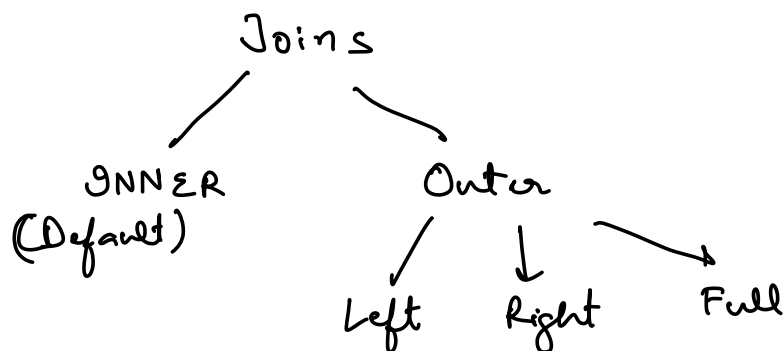
NATURAL JOINS

```

SELECT *
FROM Students ✓
NATURAL JOIN Batches ✓
  
```

- LEFT, RIGHT, Full Outer
- CROSS JOIN
- Advice on how to write joins?



Batches

id	name
①	B1
②	B2
③	B3

Students

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

Select *

FROM Batches b

→ for batch

~ for students

JOIN Students s
on b.id = s.bid

merge

b.id	b.name	s.id	s.name	sb.id
1	B1	1	A	1
1	B1	2	B	1
2	B2	4	D	2
2	B2			

★ All batches are reqd. whether or not any students are assigned to it.

SELECT ★
FROM Batches b ✓
LEFT JOIN Students s
on b.id = s.bid

1	B1	1	A	1
1	B1	2	B	1
2	B2	4	D	2
3	B3	n✓	n✓	n✓

SELECT ★
FROM Batches b
RIGHT JOIN Students s
ON b.id = s.bid

1	B1	1	A	1
1	B1	2	B	1

2	B2	4	D	2
null✓	null✓	3	C	null

Full outer Join

Not available in MySQL

1. Make an inner join

2. If any row from left side is missing, add it to results with null in right-side columns.

3. If any row from right side is missing, add it to results with null in left-side columns.

Inner Join	1	B1	1	A	1
	1	B1	2	B	1
	2	B2	4	D	2
	3	B3	n	n	n
Right	n	null	3	C	null

Left

→ Cross Join (Cartesian product)

A, B, C

A C

1 2 3
S1

1 2 3
S2

$S1 \times S2 =$

A, D	(C, D)
A, E	(C, E)
B, D	
B, E	

id	name
1	A
2	B
3	C

T1

id	name
1	D
2	E

T2

Select *
FROM T1
CROSS JOIN T2

SELECT *
FROM T1, T2

1	A	1	D
1	A	2	E
2	B	1	D
2	B	2	E
3	C	1	D
3	C	2	E

id	name
1	B1
2	B2
3	B3

Batches

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

low performance

Students

```
SELECT b.name, s.name  
FROM Batches b, Students s  
WHERE b.id = s.id
```

12 rows
9 rows
3 rows

↓

```
SELECT b.name, s.name  
FROM Batches b  
JOIN Students s  
ON b.id = s.id
```

3 rows