Agenda :

i) Joins

ii) Inner joins & self joins

iii) left joins

iv) Right joins

9th Hard day challenge :

1. Assignments + Revision
2. Backlog (Assignments of prev. session)
3. Additional Questions

Joins

- Stitches rows of two tables together.

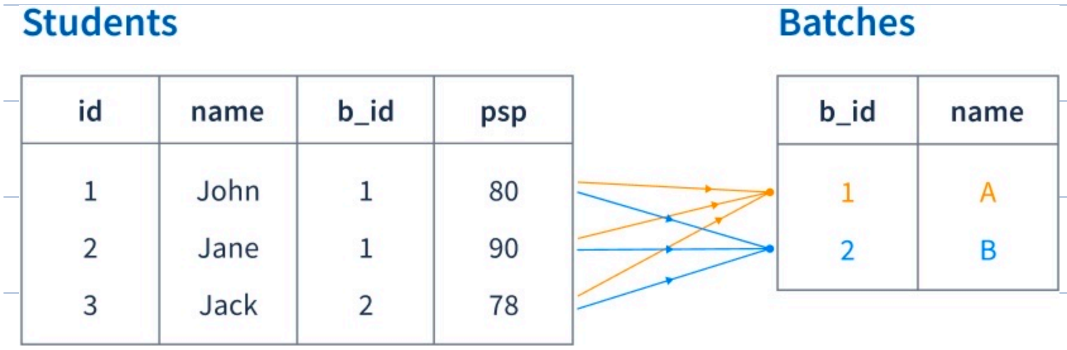
Students

id	name	b_id	psp
1	John	1	80
2	Jane	1	90
3	Jack	2	78
4	Rahul	Null	99

Batches

b_id	name
1	A
2	B
3	C

Now let's stitch the student's rows with the batch's rows in order to get the output.



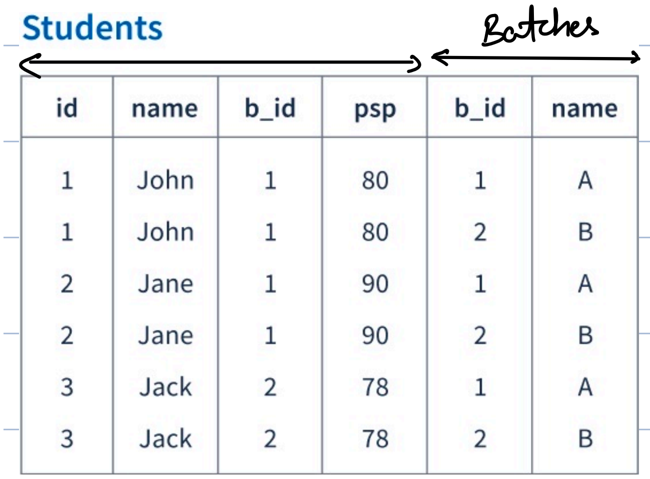
Students

id	name	b_id	psp
1	John	1	80
2	Jane	1	90
3	Jack	2	78

Batches

b_id	name
1	A
2	B

We get the following output



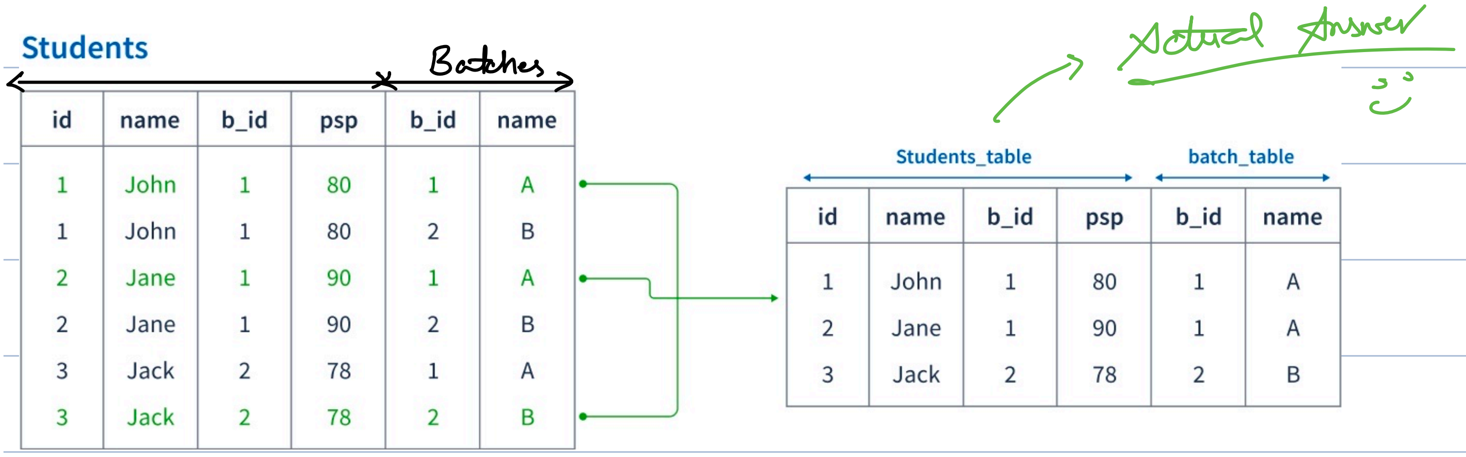
Students

Batches

id	name	b_id	psp	b_id	name
1	John	1	80	1	A
1	John	1	80	2	B
2	Jane	1	90	1	A
2	Jane	1	90	2	B
3	Jack	2	78	1	A
3	Jack	2	78	2	B

Since this isn't the desired output and we only want data which matches the b_id row of students table with the b_id row of batch table. Hence we filter out the undesired data.

After filtering we get the following final table :



Students

Batches

id	name	b_id	psp	b_id	name
1	John	1	80	1	A
1	John	1	80	2	B
2	Jane	1	90	1	A
2	Jane	1	90	2	B
3	Jack	2	78	1	A
3	Jack	2	78	2	B

Actual Answer

Students_table

batch_table

id	name	b_id	psp	b_id	name
1	John	1	80	1	A
2	Jane	1	90	1	A
3	Jack	2	78	2	B

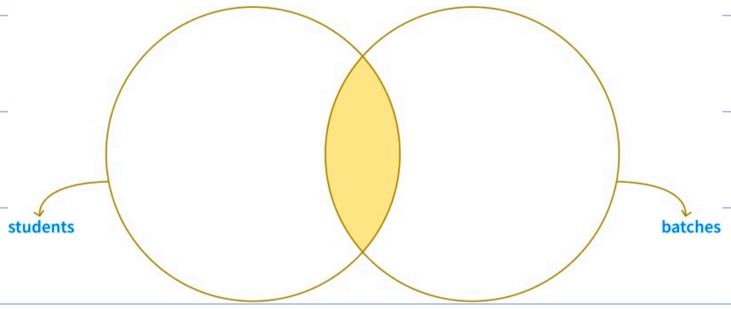
< / > Syntax

```
SELECT  students.name, batches.name

FROM students

JOIN   batches

ON     students.b_id = batches.b_id
```



Self - Joins (Joining the table with itself)

- Here we join the tables with itself.

Consider the following table :

```
select  — —

from students                ✗

where b_id = id;
```

Students

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

Question : Get name of all the students along with their buddy names.

Expected Output

name	buddy_id
A	B
B	C
C	A
D	A
E	D

' But how can we get buddy names? '

' By joining the students table with itself...'

Example

Students

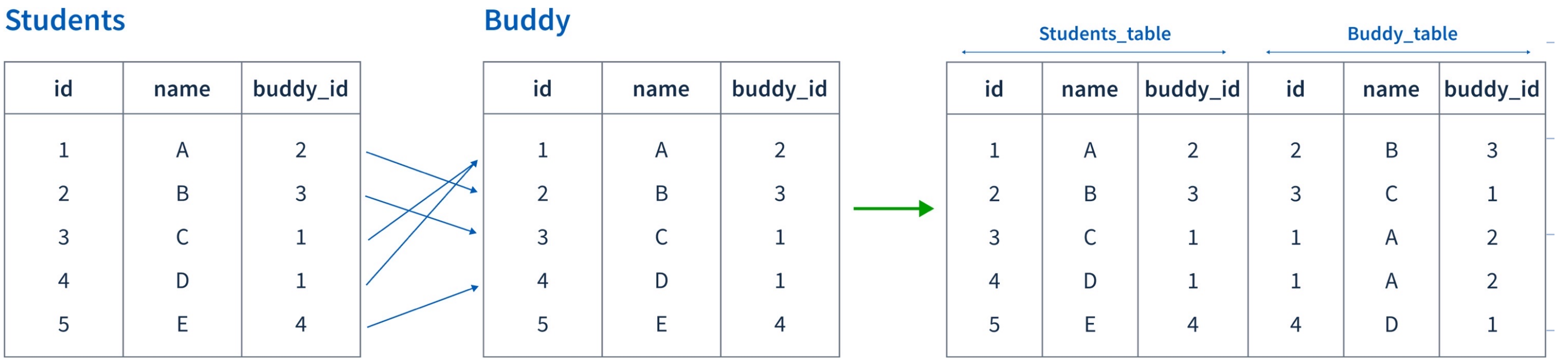
id	name	buddy_id
1	A	2
2	B	3
3	C	1
4	D	1
5	E	4

Buddy/students

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

Here we will join the given tables i.e students tables buddy_id with buddy tables id.

As shown below.



Students

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

Buddy

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

Students_table			Buddy_table		
id	name	buddy_id	id	name	buddy_id
1	A	2	2	B	3
2	B	3	3	C	1
3	C	1	1	A	2
4	D	1	1	A	2
5	E	4	4	D	1

< / > Syntax

```
SELECT  S.name, B.name
FROM Students as S
JOIN   Students as B
ON     S.buddy_id = B.id
```

2. Outer Join

It includes all the rows even if the condition doesn't match.

Types of Outer Join :

a. Left Join

b. Right Join

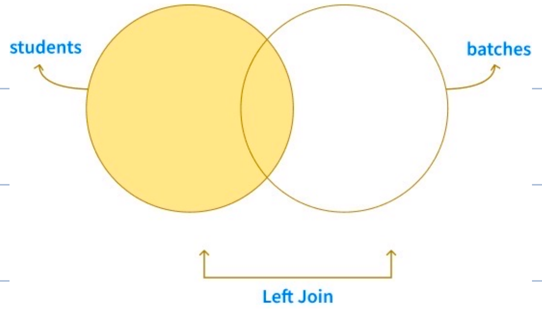
a) Left Join

- Gives all the rows from left table and only matching rows from the right table.

Example :

Get all students along with their batch_name.

Also give the data for unassigned students.



Students

id	name	b_id	psp
1	John	1	80
2	Jane	Null	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

Batches

b_id	name
1	A
2	B
3	C
4	D

Output

Students_table				batch_table	
id	name	b_id	psp	b_id	name
1	John	1	80	1	A
2	Jane	Null	90	Null	Null
3	Jim	2	85	2	B
4	Jenny	3	95	3	C
5	Jack	2	78	2	B

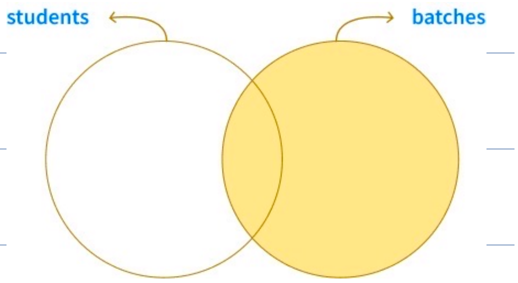
```
select *
from students s
left join batches b
on s.b_id = b.b_id ;
```

b) Right Join

- Gives all the rows from right table and only matching data from the left table.

Example :

Get all students with their assigned batches. And get all the batch names even though no students are assigned to them.



Students

id	name	b_id	psp
1	John	1	80
2	Jane	Null	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

Batches

b_id	name
1	A
2	B
3	C
4	D

Output

Students_table				batch_table	
id	name	b_id	psp	b_id	name
1	John	1	80	1	A
5	Jack	2	78	2	B
3	Jim	2	85	2	B
4	Jenny	3	95	3	C
Null	Null	Null	Null	4	D

```
select *
from students s
 right join batches b
 on s.bid = b.bid ;
```