# Joins 2

* Agenda :

1. Compound Joins
2. Types of Joins
    - Inner Join
    - Outer Join
        - Left Join
        - Right Join
        - full Join
    - Cross Join
3. Using
4. ON v/s Where
5. Implicit Join
6. Natural Join

# 14th Day : → Write a lot of joins query (10-15).
→ Solve all assignment questions.

## * **Joining multiple tables :**

students

| id | name | Instructor_id | b_id |
|---|---|---|---|
| 1 | Jim | 1 | 2 |
| 2 | Jenny | 1 | 1 |

Instructors

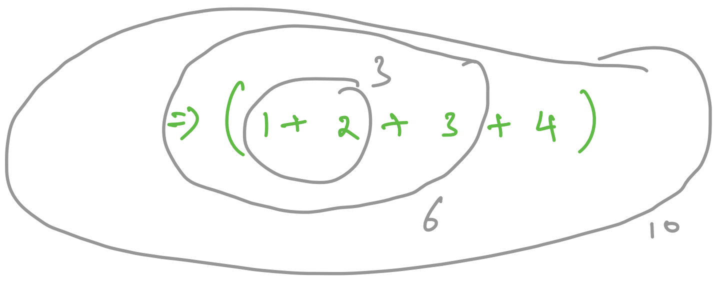
| id | name |
|---|---|
| 1 | Rahul |
| 2 | Prateek |

batches

| b_id | batch_name |
|---|---|
| 1 | A |
| 2 | B |

**Q.** for every student, give their corresponding instructor name & batch name.

| name | Instructor_name | batch_name |
|---|---|---|
| Jim | Rahul | B |



```
select    *
from   students    s
 join   batches    b
 on    s.batch_id  =  b.batch_id
 join    instructors  i
  on     s.instructor_id  =  i.id
```

intermediatory table

combined with instructors

* Compound Joins :

-> for every film, name the films which were released 2 years before current film & 2 years after current film.

| name | release_year | rental_rate |
|---|---|---|
| Hera Pheri | 2008 | 2 |
| Robot | 2009 ✓ | 3 |
| Welcome | 2011 | 4 |

2006

2010

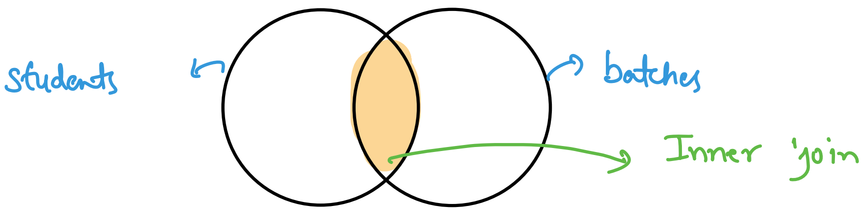
```
select
from    film    f1
join    film    f2
 on     f2.release_year  between  f1.year + 2  And  f1.year - 2
            and   f2.rental_rate  >  f1.rental_rate
```

# In compound joins we have multiple condtn on multiple coln.

## Type of Joins :

```
select   *
from     students   s
join     batches    b
 On      s.id  =  b.id ;
```

# This type of join is called inner join.



# Inner join doesn't include row where condition is not matched.

PS : ( Join) → inner Join

inner Join ↗

---

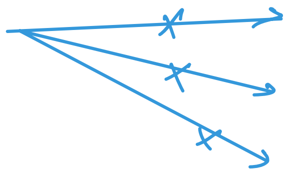
* **Outer Joins :**

→ Outer Join includes all rows, even though they might not match conditions.

1. Left Join ( left outer join)
2. Right Join ( Right outer join)
3. full Join ( full outer join)

Students

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

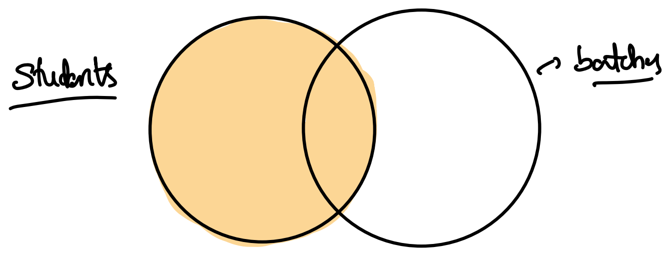


Batches

| b_id | name |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |

# In inner join we will miss upon John's data

⇒ left Join :

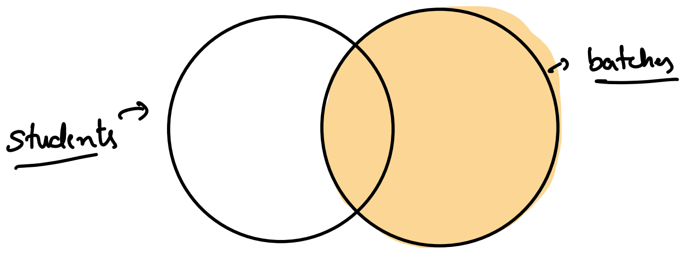


```
select  *
from    students s
left    join batches b
on      s.batch_id = b.batch_id ;
```

⇒ It gives all data which matches condition

⇒ It gives all data from left table for which condⁿ doesn't match. for corresponding fields in batches it fills Null values.

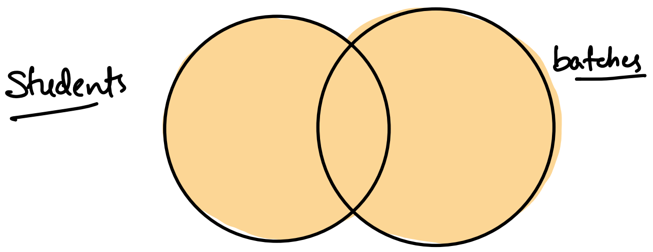
## * Right Join :



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

# Gives all data for which condtn maches.
# Gives all right table (batches) data even though for which condtn doesn't match.
# for empty fields on left side fill null.

## * full Join



```
select *
from students s
full join batches b
on s.id = b.id ;
```

No. of rows ⇒ Students + batches

$\longrightarrow$

* **Cross Join :**

```
select *
from students s
cross join batches b ;
```

# Cross Join gives us all possible combinations of left & right table

Colors

| id | Colors |
|---|---|
| 1 | Red |
| 2 | Black |

M

Sizes

| id | Size |
|---|---|
| 1 | M |
| 2 | L |

N

| id | color | id | size |
|---|---|---|---|
| 1 | Red | 1 | M |
| 1 | Red | 2 | L |
| 2 | Black | 1 | M |
| 2 | Black | 2 | L |

# No. of row (total) = M x N

---

* **Natural Joins :**

| id | name | phn_no |
|---|---|---|
| | | |

| id | phn_no. |
|---|---|
| | |

# If I want to join these tables on id as well as phn_no.

```
select  *
from  table1
Natural  join  table2
```

# Joins table1 & table2 using colm name which are same across both tables.

---

* <u>Join using</u> <u>ON v/s WHERE</u> ?

⇒ If we will use 'where' in join statement instead of 'ON' then the join will naturally act as a 'Cross Join'.

⇒ Where will work, but it will be inefficient.

⇒ Writing queries with 'ON' are faster.

---

| | |
|---|---|
| students | id \| name |
| employees | id \| name |
| investors | id \| name |

```
select name from students
Union
Select name from investors;
```