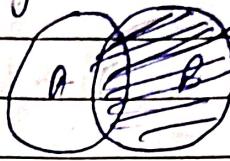
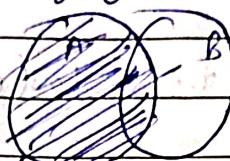
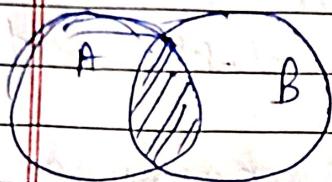


## Joins in SQL

Inner join    Left join    Right join    full join



Outer Joins

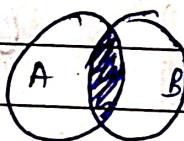
Joins are used to combine rows from two or more tables based on related column between the tables

Used to find common info in two tables

Not compulsory to have foreign key

### Inner Join

Returns records that have matching values in both tables



### Syntax

SELECT column(s)  
FROM TableA

INNER JOIN TableB

ON TableA.col\_name = TableB.col\_name;

ex

Table1 → student

Date / /  
Page No.

table 2 → course

stu-id	Name
102	Armaan
103	Vinod
104	Vishal

stu-id	course
102	english
103	Hindi
104	Maths
107	Science

SELECT \*

FROM student

INNER JOIN course

ON ~~student~~ student.stu-id = course.stu-id ;

O/p

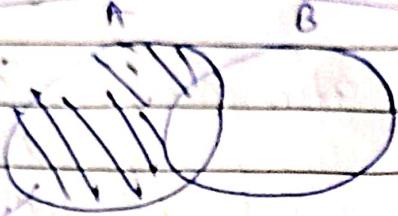
stu-id	name	stu-id	course
102	Armaan	102	English
103	Vinod	103	Hindi
104	Vishal	104	Maths

NOTE

alias → used to make big table names in short

ex → SELECT \*  
FROM student AS s

## Left Exclusive Join



BII IS NULL

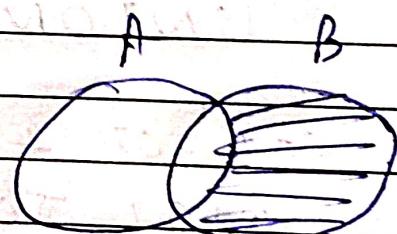
Gives only the Data of A table  
(which does not overlap with  
B table)

```
SELECT *
FROM Student
LEFT JOIN course
ON student.stu_id = course.cou_id
WHERE course.stu_id IS NULL;
```

here we use LEFT Join and select  
only those values where table B  
is NULL.

## Right Exclusive Join

Given only data of B table  
not common to both.

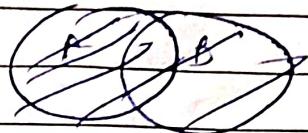


```
SELECT *
FROM Student AS a
RIGHT JOIN course AS b
ON a.student_id = b.stu_id
WHERE a.stu_id IS NULL;
```

Output →

stu_id	name	stu_id	course
102	Aman	102	English
103	Vinod	103	Hindi
104	Vishal	104	Maths
null	null	107	Science

## FULL JOIN



⇒ Returns all records when there is a match in either left or right table

use UNION of LEFT and RIGHT Join

SELECT \* FROM student  
LEFT JOIN course  
ON student.stu\_id = course.stu\_id

UNION

(left join  
UNION  
Right Join)

SELECT \* FROM student  
RIGHT JOIN course  
ON student.stu\_id = course.stu\_id;

stu_id	name	stu_id	course
102	Aman	102	English
103	Vinod	103	Hindi
104	Vishal	104	Maths
null	null	107	Science

## II LEFT JOIN

~~SELECT column(s)  
FROM table1-name~~

LEFT JOIN table2-name,

ON table1.col-name = table2.col-name;

Returns all records from table 1 and matched records from table 2.

ex,

~~SELECT \*  
FROM student as s  
LEFT JOIN course as c  
ON s.stu-id = c.stu-id~~

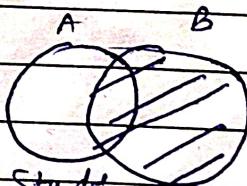
O/p

Stu-id	name	Stu-id	course
102	Aman	102	English
103	Vinod	103	Maths
104	Vishal	104	Maths

## III Right JOIN

Gives the data of right table and the data common to both tables.

~~SELECT \*  
FROM student  
RIGHT JOIN course  
ON s.stu-id = c.stu-id~~



student course

Ex -> get names of all students from student table who got more than average marks.

Step 1 -> find avg of class marks

Step 2. find names of students with marks > avg

Ans ->

```
SELECT avg(marks) } o/p = 87.66  
FROM student
```

```
SELECT names  
FROM student  
WHERE marks > 87.66 ;
```

Now to combine these two queries

2) SELECT names  
FROM student  
WHERE marks > (SELECT avg(marks)  
FROM student);  
↳ (dynamic)

gives right result when there are updates in table.

Ex -> SELECT all students with even numbers;

Step 1, find all even numbers.

Step 2. find names with even no.

## UNION

- Used to combine the result set of two or more SELECT statements

- Gives UNIQUE records

To use it,

- every SELECT should have same no. of columns
- columns must have similar data types
- columns in every SELECT should be in same order.

### Syntax :

SELECT column(s) FROM table A

UNION

SELECT column(s) FROM table B

UNION ALL gives all values even duplicates also

### SQL Sub-Queries

A query with more than one SQL query.

( nested query ) on Inner query

It involves 2 SELECT statements

Syntax :-

SELECT column(s)  
FROM Table-name ;

WHERE

operator

( subquery ) ;

Ex → Student

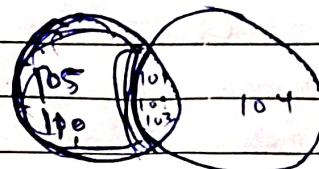
stu-id	name
101	Ram
102	Shyan
103	Praveen
105	Naman
110	Rakesh

Course

stu-id	course-name
101	English
102	Hindi
103	Maths
104	Science

Op →

stu-id	name	Null	Null
105	name	Null	Null
110	Rakesh	Null	Null



Student Course

stu-id	name	stu-id	course-name
null	null	104	science

## SELF JOIN

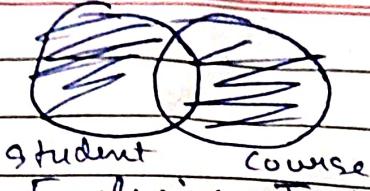
⇒ Regular join but it joins a table with itself.

```
SELECT column(s)
  FROM table A as a
  JOIN tableA as b
  ON A.col-name = B.col-name
```

⇒

A

B



student                  course

⇒ Exclusive JOIN query to SELECT

only unique data from both tables.

⇒

SELECT \* FROM student AS A

LEFT JOIN course AS B

ON A.stu-id = B.stu-id

WHERE B.stu-id IS NULL

UNION

UNION

SELECT \* FROM student AS A

RIGHT JOIN course AS B

ON A.stu-id = B.stu-id

WHERE A.stu-id IS NULL;

## SQL Sub Queries with SELECT

`SELECT (SELECT ...)`

not used much in SQL.

### (MySQL views)

Used to view or show only some part of the total data and perform queries on it.

e.g. from customer data, the accounts team only need to see bank details from the table.

⇒ `CREATE VIEW view1 AS`

`SELECT card-no, name FROM customer;`

⇒ now run queries on the sub-data

`SELECT * FROM view1;`

To drop

`DROP VIEW view1;`

Ans) SELECT marks

FROM student

WHERE roll\_no IN (

SELECT roll\_no

FROM student

WHERE

roll\_no % 2 = 0 );

# Sub queries in FROM

Q) - find the max marks from the students of Delhi

Step 1. find students of Delhi

Step 2. find max marks using the subquery in step 1

Ans  
→

SELECT →

SELECT max(marks)

FROM ( SELECT \*

FROM student

WHERE city = "Delhi" )

AS Temp ;

it is mandatory to use alias here.

(Q2)

SELECT max(marks) FROM student

WHERE

city = "Delhi" ;

• Coalesce ( null , 5 , 10 )

Returns 5

It returns the first non null or 0 value in the list.

It is used to show zero at place of NULL is the table's

• CEIL → ( )

Rounds up the result to the next integer

• Replace ( )

Used to replace any word or letters

ex → REPLACE ( salary , '0' , ' ') + 0 }

2002 → 22