LET'S START WITH SQL :)

Joins in SQL

Joins are used to combine rows from two or more tables based on a related or shared or common column between them. There are commonly 4 types of joins including **INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, SELF JOIN , CROSS JOIN.**

id	Name	Age
1	Riya	17
2	Rahul	18
3	Ram	17

Student

id	course_id	course_name
1	101	Eng
2	102	Hin
3	103	PhE

Course

LET'S START WITH SQL :)

Joins in SQL

Q. Is Foreign Key important for performing joins?

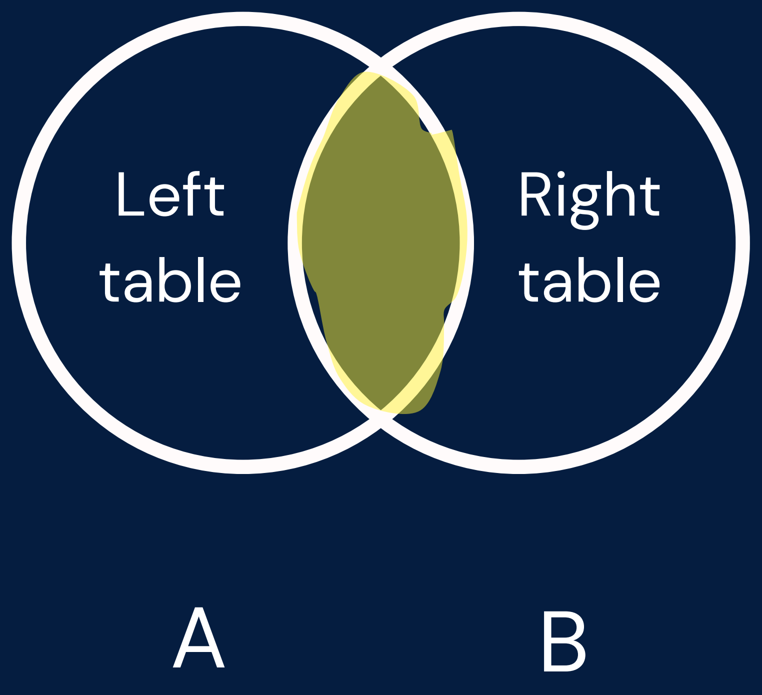
->Joins can be performed based on any columns that establish a relationship between tables, not just FK constraints, so its not necessary.

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

1. Inner Join



rollno	name
1	Ram
2	Rahul
3	Riti

Student

rollno	**c_name**
2	Hindi
3	Eng
4	Maths

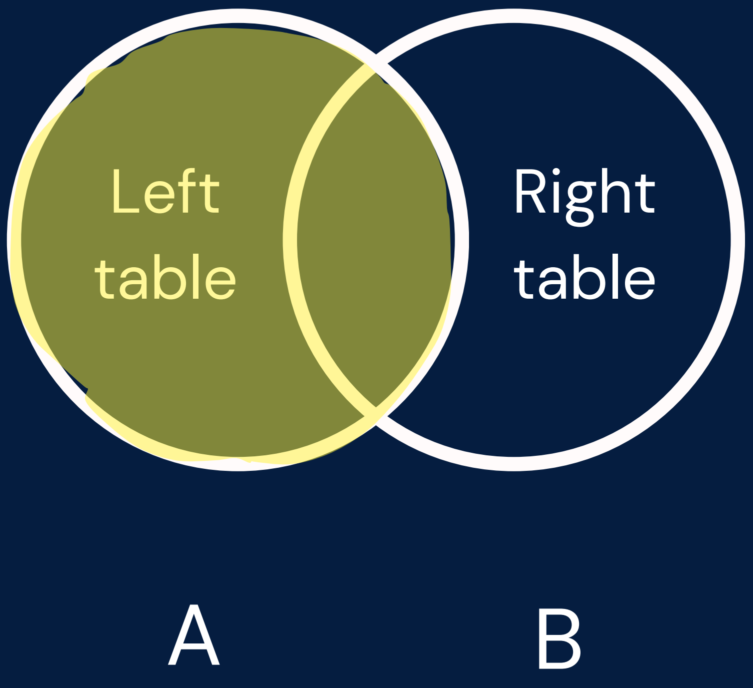
Course

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

2. Left Join/Left Outer Join



rollno	name
1	Ram
2	Rahul
3	Riti

Student

rollno	**c_name**
2	Hindi
3	Eng
4	Maths

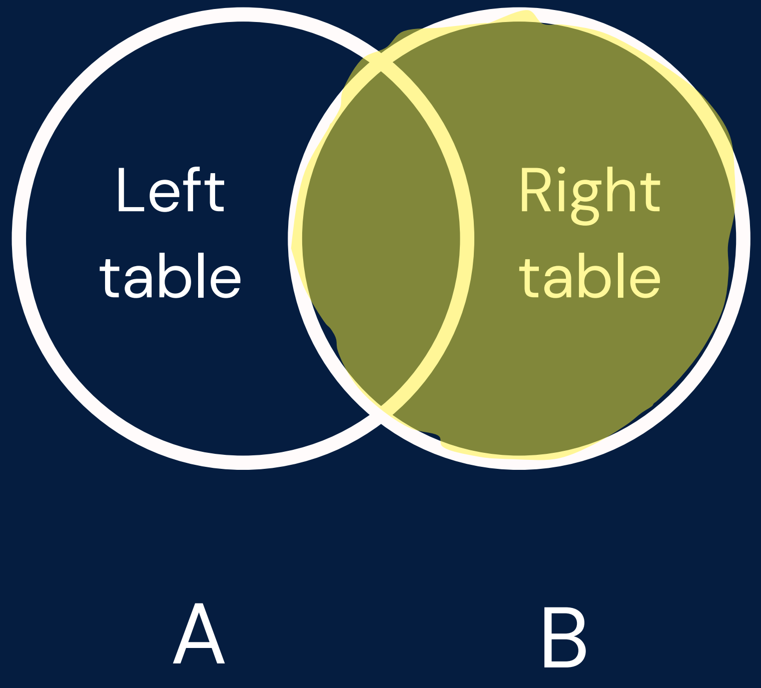
Course

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

3. Right Join/ Right Outer Join



rollno	name
1	Ram
2	Rahul
3	Riti

Student

rollno	c_name
2	Hindi
3	Eng
4	Maths

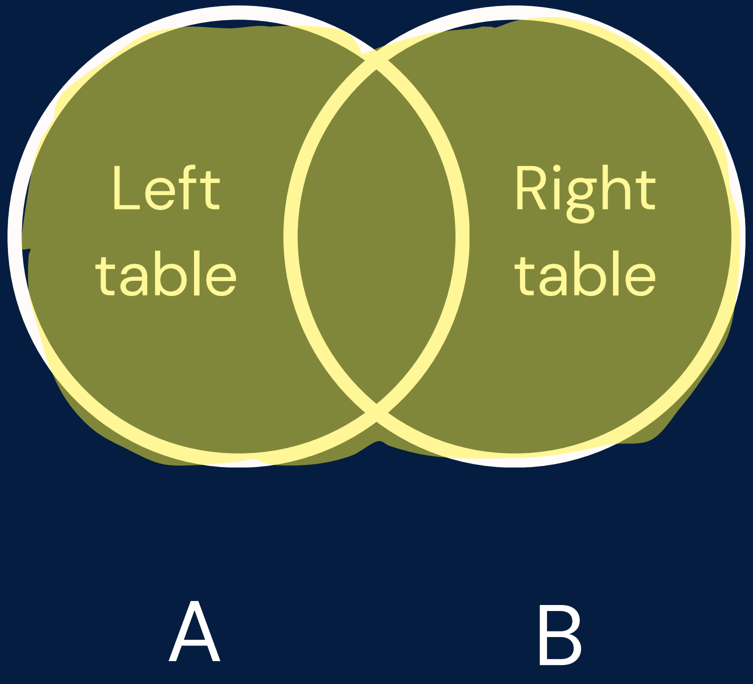
Course

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

4. Full Join/Full Outer Join



rollno	name
1	Ram
2	Rahul
3	Riti

Student

rollno	c_name
2	Hindi
3	Eng
4	Maths

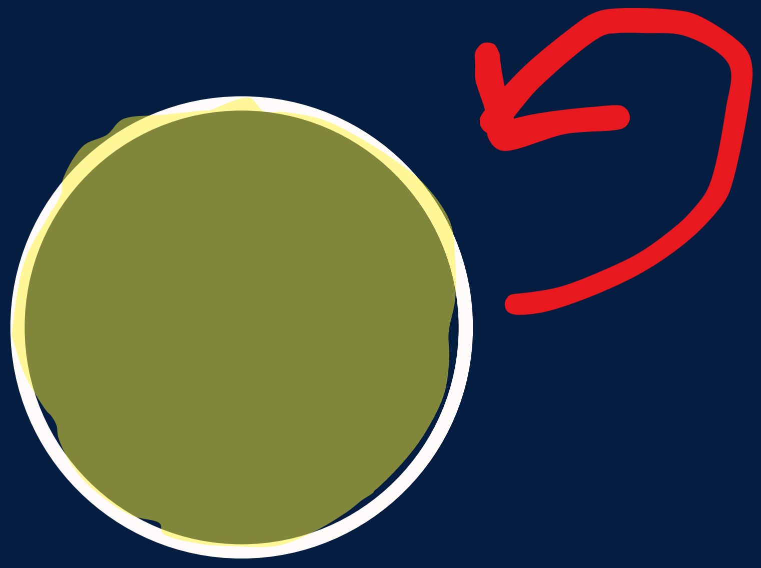
Course

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

5. Self Join



A

rollno	name
1	Ram
2	Rahul
3	Riti

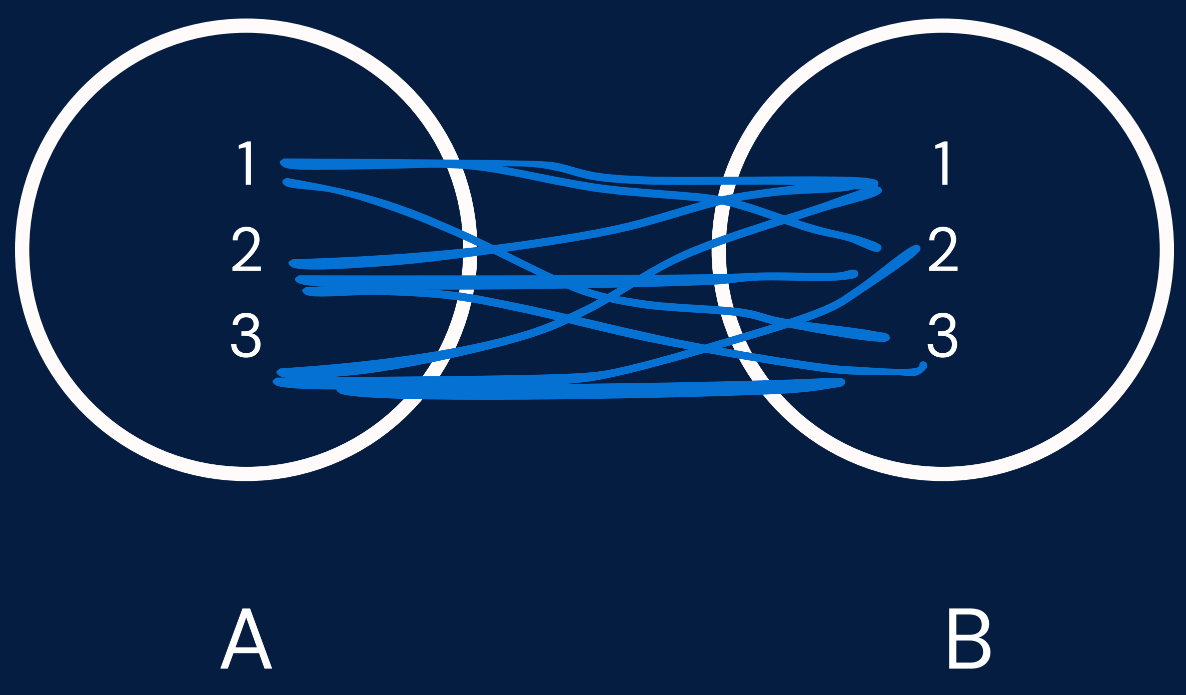
Student

LET'S START WITH SQL :)

Joins in SQL

Types of Joins in SQL

6. Cross Join



rollno	name
1	Ram
2	Rahul
3	Riti

Student

rollno	**c_name**
2	Hindi
3	Eng
4	Maths

Course

LET'S START WITH SQL :)

Joins in SQL

2. Left Join/Left Outer Join : It is used to fetch all the records from the left table along with matched records from the right table.
If there are no matching records in the right table, NULL values are returned for the columns of the right table.

Query:

```
SELECT columns
FROM table1
LEFT JOIN table2
ON table1.colName = table2.colName;
```

Left table : the table specified before the LEFT JOIN keyword
Right table : the table specified after the LEFT JOIN keyword

LET'S START WITH SQL :)

Joins in SQL

Query:

```
SELECT *
FROM customer
LEFT JOIN order
ON customer.id = order.id;
```

id	**name**	**id**	**o_name**
101	**Ram**	**null**	**null**
102	Rahul	102	Fruit
103	Riti	103	Ball

id	name
101	Ram
102	Rahul
103	Riti

customer

id	**o_name**
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Joins in SQL

3. Right Join/ Right Outer Join : It is used to fetch all the records from the right table along with matched records from the left table.

If there are no matching records in the left table, NULL values are returned for the columns of the left table.

Query:

```
SELECT columns
FROM table1
RIGHT JOIN table2
ON table1.colName = table2.colName;
```

Left table : the table specified before the RIGHT JOIN keyword

Right table : the table specified after the RIGHT JOIN keyword

LET'S START WITH SQL :)

Joins in SQL

Query:

```
SELECT *
FROM customer
RIGHT JOIN order
ON customer.id = order.id;
```

id	o_name	id	name
102	Fruit	102	Rahul
103	Ball	103	Riti
104	utensils	null	null

id	name
101	Ram
102	Rahul
103	Riti

customer

id	o_name
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Joins in SQL

1. Inner Join : It helps us in getting the rows that have matching values in both tables, according to the given join condition.

Query:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.colName = table2.colName;
```

id	name
101	Ram
102	Rahul
103	Riti

Customer

id	**o_name**
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Joins in SQL

Query: It only returns rows where there is a matching id in both tables

```
SELECT *
FROM customer
INNER JOIN order
ON customer.id = order.id;
```

id	name	id	o_name
102	Rahul	102	Fruit
103	Riti	103	Ball

id	name
101	Ram
102	Rahul
103	Riti

customer

id	o_name
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Joins in SQL

4. Full Join/Full Outer Join: It returns the matching rows of both left and right table and also includes all rows from both tables even if they don't have matching rows.

If there is no match, NULL values are returned for the columns of the missing table.

In MySQL, the syntax for a full join is different compared to other SQL databases like PostgreSQL or SQL Server.

MySQL does not support the FULL JOIN keyword directly. So we use a combination of LEFT JOIN, RIGHT JOIN, and UNION to achieve the result.

LET'S START WITH SQL :)

Joins in SQL

4. Full Join/Full Outer Join:

Query:

```
SELECT columns
FROM table1
LEFT JOIN table2
ON table1.colName = table2.colName;

UNION

SELECT columns
FROM table1
RIGHT JOIN table2
ON table1.colName = table2.colName;
```

LET'S START WITH SQL :)

Joins in SQL

Query:

```
SELECT *
FROM customer
LEFT JOIN order
ON customer.id = order.id;
UNION
SELECT *
FROM customer
RIGHT JOIN order
ON customer.id = order.id;
```

id	name
101	Ram
102	Rahul
103	Riti

customer

id	**o_name**
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Joins in SQL

Result :

id	**name**	**id**	**o_name**
101	Ram	null	null
102	Rahul	102	Fruit
103	Riti	103	Ball
null	null	104	Utensils

LET'S START WITH SQL :)

Joins in SQL

6. CrossJoin: It combines each row of the first table with every row of the second table.

Query:

```
SELECT *
FROM table1
CROSS JOIN table2;
```

id	name
101	Ram
102	Rahul

Customer

o_id	o_name
1	Fruit
2	Ball

Order

It results in a new table where the number of rows is equal to the product of the number of rows in each table. (m*n)

LET'S START WITH SQL :)

Joins in SQL

Result :

id	name	o_id	o_name
101	Ram	1	Fruit
101	Ram	2	Ball
102	Rahul	1	Fruit
102	Rahu	2	Ball

LET'S START WITH SQL :)

Joins in SQL

5. Self Join: A self join in SQL is a type of join where a table is joined with itself. It is a type of inner join.

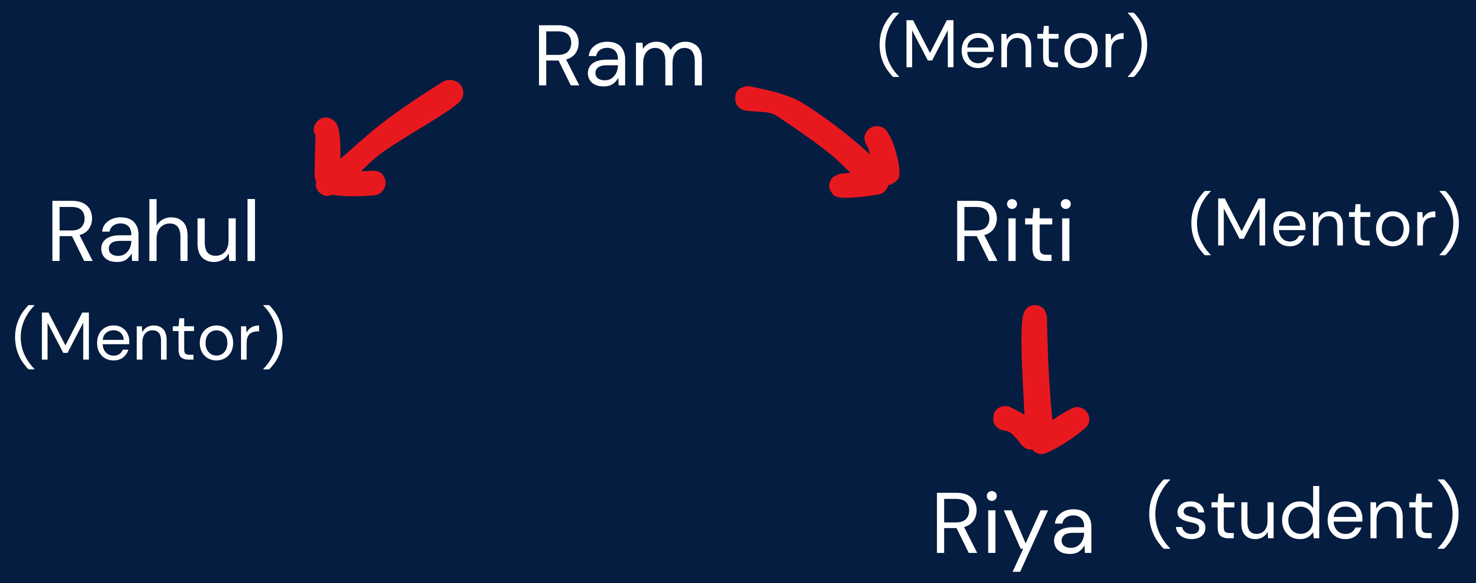
Query:

```
SELECT columns
FROM table as t1
JOIN table as t2
ON t1.colName = t2.colName
```

t1 and t2 are aliases for the table, used to distinguish between the order rows.

LET'S START WITH SQL :)

Joins in SQL



Ram (Mentor)

Rahul (Mentor)

Riti (Mentor)

Riya (student)

Query :

```
SELECT s1.name as mentor_name, s2.name as name
FROM student as s1
JOIN student as s2
WHERE s1.s_id=s2.mentor_id
```

s_id	**name**	**mentor_id**
1	Ram	null
2	Rahul	1
3	Riti	1
4	Riya	3

LET'S START WITH SQL :)

Joins in SQL

Result :

mentor_name	**name**
Ram	Riti
Ram	Rahul
Riti	Riya

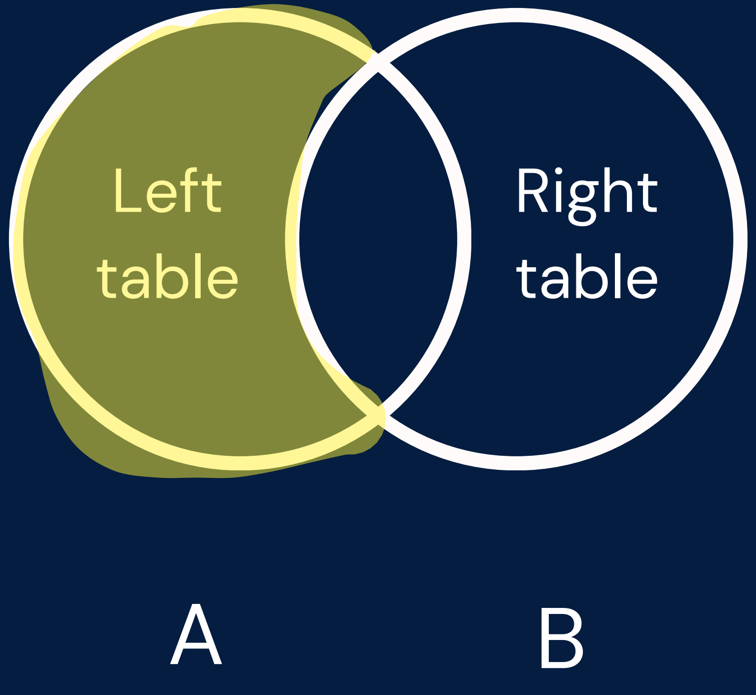
LET'S START WITH SQL :)

Exclusive Joins in SQL

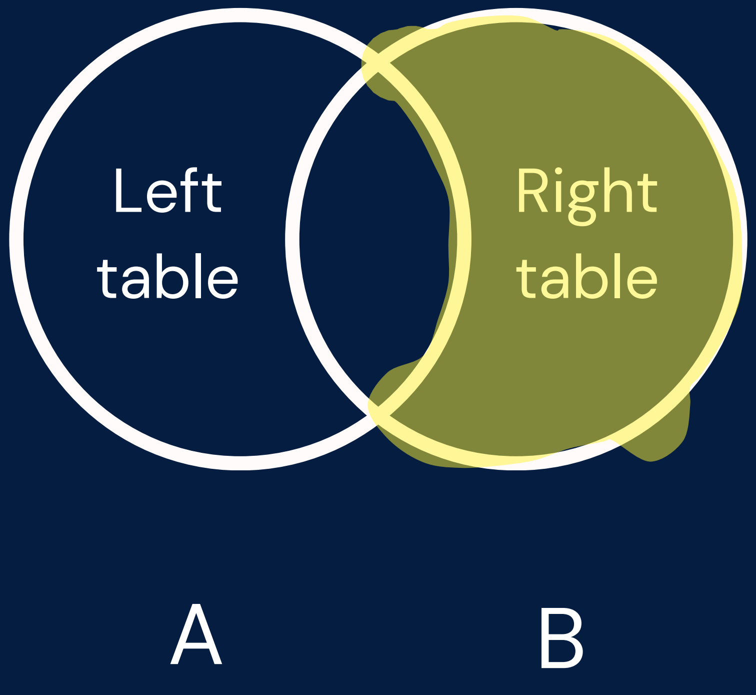
Exclusive joins are used when we want to retrieve data from two tables excluding matched rows. They are a part of outer joins or full outer join.

Types :

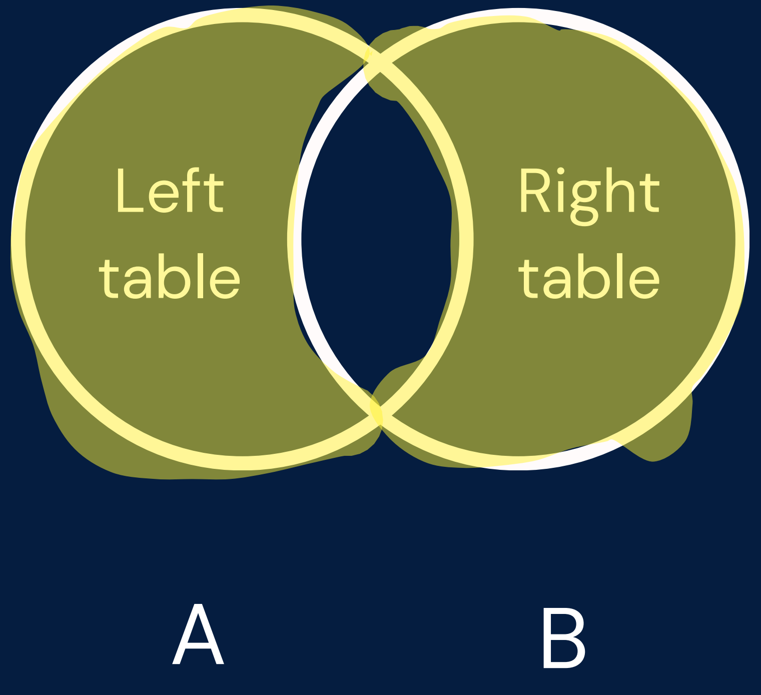
1.Left Exclusive JOIN



2.Right Exclusive JOIN



3.Full Exclusive JOIN



LET’S START WITH SQL :)

Exclusive Joins in SQL

Left Exclusive JOIN: When we retrive records from the left table excluding the ones matching in both left and right table .

Query:

```
SELECT columns
FROM table1
LEFT JOIN table2
ON table1.colName = table2.colName;
WHERE table2.colName IS NULL;
```

id	name
101	Ram
102	Rahul
103	Riti

customer

id	**o_name**
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Exclusive Joins in SQL

Right Exclusive JOIN: When we retrive records from the right table excluding the ones matching in both left and right table .

Query:

```
SELECT columns
FROM table1
RIGHT JOIN table2
ON table1.colName = table2.colName;
WHERE table1.colName IS NULL;
```

id	name
101	Ram
102	Rahul
103	Riti

customer

id	o_name
102	Fruit
103	Ball
104	Utensils

Order

LET'S START WITH SQL :)

Exclusive Joins in SQL

Full Exclusive JOIN: When we retrive records from the right table and left table excluding the ones matching in both left and right table .

Query:

```
SELECT columns
FROM table1
LEFT JOIN table2
ON table1.colName = table2.colName;
WHERE table2.colName IS NULL;
UNION
SELECT columns
FROM table1
RIGHT JOIN table2
ON table1.colName = table2.colName;
WHERE table1.colName IS NULL;
```