

# What are SQL joins

20 February 2023 15:28

In SQL (Structured Query Language), a join is a way to combine data from two or more database tables based on a related column between them.

Joins are used when we want to query information that is distributed across multiple tables in a database, and the information we need is not contained in a single table. By joining tables together, we can create a virtual table that contains all of the information we need for our query.

## But why have data in multiple tables?

20 February 2023 15:28

user_id	name	state	city
1	Bharat	Gujarat	Ahmedabad
2	Pearl	Maharashtra	Pune
3	Jahan	Madhya Pradesh	Bhopal
4	Divsha	Rajasthan	Jaipur
5	Kasheen	West Bengal	Kolkata
6	Hazel	Karnataka	Bangalore
7	Sonakshi	Jammu and Kashmir	Kashmir

man

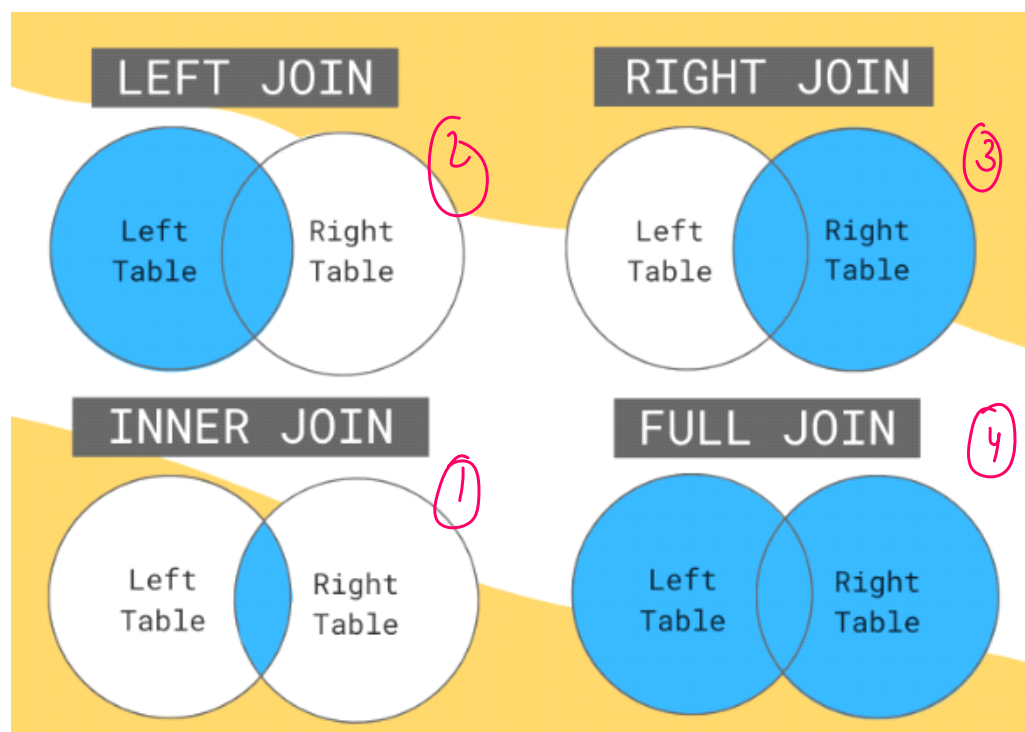
order_id	user_id	order_date
B-25601	1	01-04-2018
B-26011	1	12-02-2019
B-26074	1	21-03-2019
B-25602	2	01-04-2018
B-26012	2	13-02-2013
B-26075	2	21-03-2019
B-25603	3	03-04-2018
B-26013	3	13-02-2019

userid	name	state	city
1	Bha	Guj	Ah
1	Bha	Guj	Ah
1	Bha	Guj	Ah

$\left\{ \begin{array}{l} 08 \\ B-256 \\ B-260 \\ B-25 \end{array} \right\}$  ,  $\left\{ \begin{array}{l} n \\ Bharat \\ Shm \\ Jn \end{array} \right\}$

# Types of Joins

20 February 2023 15:30



{ Cross join  
self join }

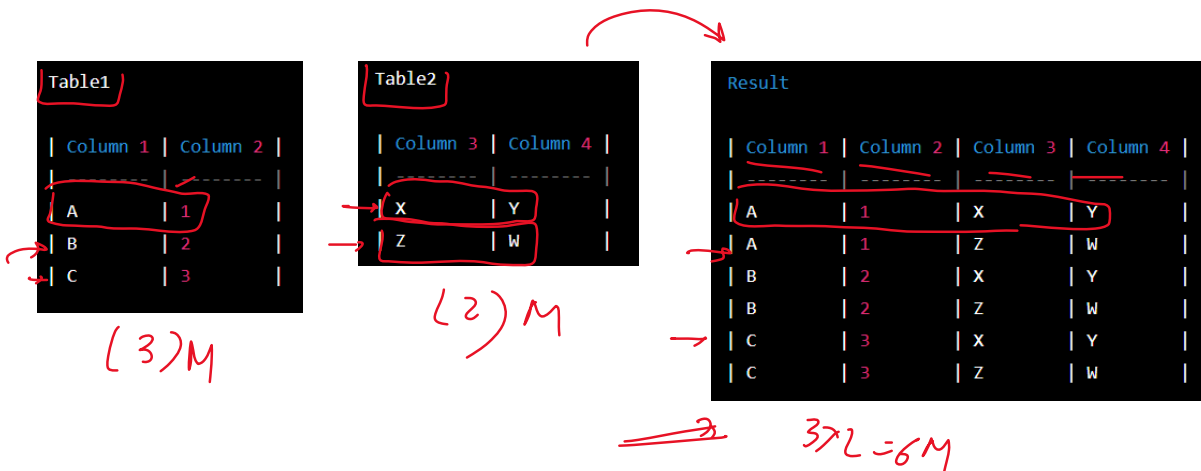
## Cross Joins -> Cartesian Products

20 February 2023 15:29

In SQL, a cross join (also known as a Cartesian product) is a type of join that returns the Cartesian product of the two tables being joined. In other words, it returns all possible combinations of rows from the two tables.

Cross joins are not commonly used in practice, but they can be useful in certain scenarios, such as generating test data or exploring all possible combinations of items in a product catalogue. However, it's important to be cautious when using cross joins with large tables, as they can generate a very large result set, which can be resource-intensive and slow to process.

$$A = \{ \underline{1}, \underline{2} \}$$
$$B = \{ \underline{3}, \underline{4} \}$$
$$\underline{(1,3)} \quad \underline{(1,4)} \quad \underline{(2,3)} \quad \underline{(2,4)}$$

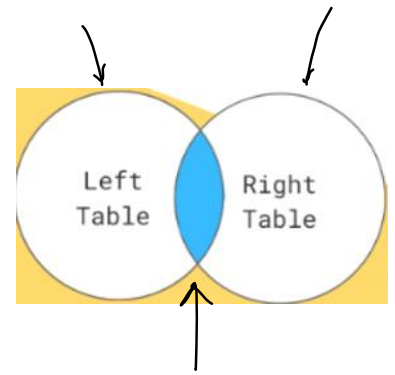


## Inner Joins

20 February 2023 15:30

In SQL, an inner join is a type of join operation that combines data from two or more tables based on a specified condition. The inner join returns only the rows from both tables that satisfy the specified condition, i.e., the matching rows.

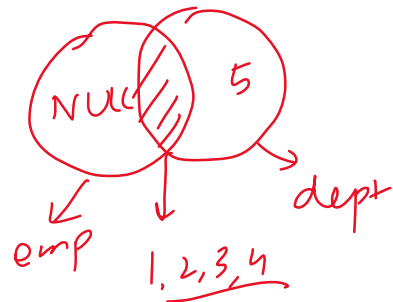
When you perform an inner join on two tables, the result set will only contain rows where there is a match between the joining columns in both tables. If there is no match, then the row will not be included in the result set.



*emp* *dept*

Employee ID	Name	Department ID	Salary
1	John Smith	1	100000
2	Jane Doe	2	50000
3	Bob Johnson	3	75000
4	Lisa Wong	1	90000
5	Mike Lee	2	120000
6	Tim Davis	4	60000
7	Sarah Chen	NULL	80000

Department ID	Department Name
1	Engineering
2	Sales
3	Finance
4	Marketing
5	Operations



*emp* *dept*

Employee ID	Name	Department ID	Salary	Department ID	Department
1	John Smith	1	100000	1	Engineering
4	Lisa Wong	1	90000	1	Engineering
2	Jane Doe	2	50000	2	Sales
5	Mike Lee	2	120000	2	Sales
3	Bob Johnson	3	75000	3	Finance

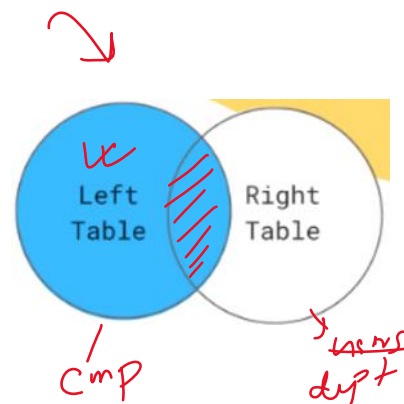
*Tim Davis*

## Left Join

20 February 2023 15:30

A left join, also known as a left outer join, is a type of SQL join operation that returns all the rows from the left table (also known as the "first" table) and matching rows from the right table (also known as the "second" table). If there are no matching rows in the right table, the result will contain NULL values in the columns that come from the right table.

In other words, a left join combines the rows from both tables based on a common column, but it also includes all the rows from the left table, even if there are no matches in the right table. This is useful when you want to include all the records from the first table, but only some records from the second table.



emp

Employee ID	Name	Department ID	Salary
1	John Smith	1	100000
2	Jane Doe	2	50000
3	Bob Johnson	3	75000
4	Lisa Wong	1	90000
5	Mike Lee	2	120000
6	Tim Davis	4	60000
7	Sarah Brown	5	80000
8	Mark Wilson	2	95000

dept

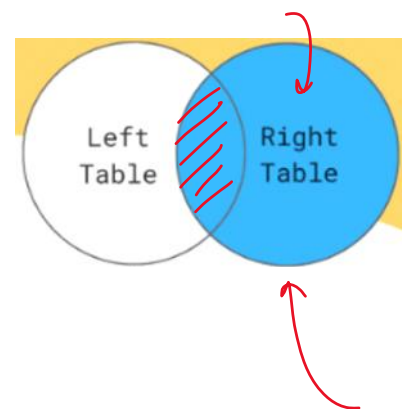
Department ID	Department Name
1	Engineering
2	Sales
3	Finance

Employee ID	Name	Department ID	Salary	Department ID	Department
1	John Smith	1	100000	1	Engineering
2	Jane Doe	2	50000	2	Sales
3	Bob Johnson	3	75000	3	Finance
4	Lisa Wong	1	90000	1	Engineering
5	Mike Lee	2	120000	2	Sales
6	Tim Davis	4	60000	NULL	NULL
7	Sarah Brown	5	80000	NULL	NULL
8	Mark Wilson	2	95000	2	Sales

# Right Join

20 February 2023 15:31

A right join, also known as a right outer join, is a type of join operation in SQL that returns all the rows from the right table and matching rows from the left table. If there are no matches in the left table, the result will still contain all the rows from the right table, with NULL values for the columns from the left table.



*emp*

Employee ID	Name	Department ID	Salary
1	John Smith	1	100000
2	Jane Doe	2	50000
3	Bob Johnson	3	75000
4	Lisa Wong	1	90000
5	Mike Lee	2	120000
7	Sarah Brown	NULL	80000
8	Mark Wilson	2	95000

*dept*

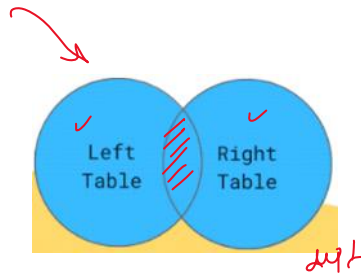
Department ID	Department Name
1	Engineering
2	Sales
3	Finance
4	Marketing
5	HR

Employee ID	Name	Department ID	Salary	Department Name
1	John Smith	1	100000	Engineering
4	Lisa Wong	1	90000	Engineering
2	Jane Doe	2	50000	Sales
5	Mike Lee	2	120000	Sales
8	Mark Wilson	2	95000	Sales
3	Bob Johnson	3	75000	Finance
NULL	NULL	4	NULL	Marketing
NULL	NULL	5	NULL	HR

# Full Outer Join

20 February 2023 15:31

A full outer join, sometimes called a full join, is a type of join operation in SQL that returns all matching rows from both the left and right tables, as well as any non-matching rows from either table. In other words, a full outer join returns all the rows from both tables and matches rows with common values in the specified columns, and fills in NULL values for columns where there is no match.



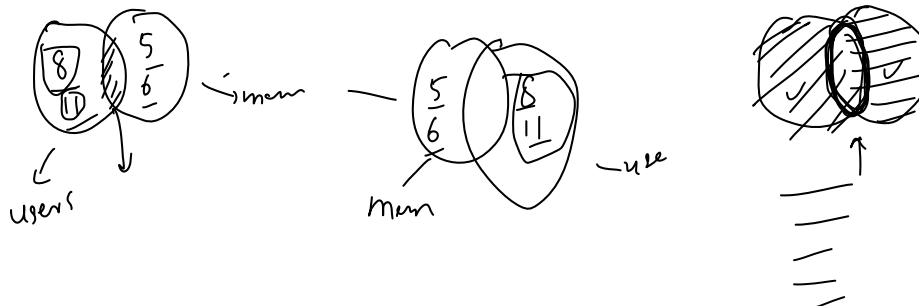
emp

emp_id	emp_name	dept_id
1	Alice	1
2	Bob	1
3	Charlie	2
4	Dave	null
5	Eve	3

dept

dept_id	dept_name
1	Sales
2	Marketing
3	Finance
4	IT
5	HR

emp_id	emp_name	dept_id	dept_id	dept_name
1	Alice	1	1	Sales
2	Bob	1	1	Sales
3	Charlie	2	2	Marketing
4	Dave	null	null	null
5	Eve	3	3	Finance
null	null	null	4	IT
null	null	null	5	HR





# SQL Set Operations

20 February 2023 15:35

1. **UNION**: The UNION operator is used to combine the results of two or more SELECT statements into a single result set. The UNION operator removes duplicate rows between the various SELECT statements.
2. **UNION ALL**: The UNION ALL operator is similar to the UNION operator, but it does not remove duplicate rows from the result set.
3. **INTERSECT**: The INTERSECT operator returns only the rows that appear in both result sets of two SELECT statements.
4. **EXCEPT**: The EXCEPT or MINUS operator returns only the distinct rows that appear in the first result set but not in the second result set of two SELECT statements.

person1      except

id	name
1	Alice
2	Bob
3	Charlie

person2

id	name
3	Charlie
4	David
5	Emily

union

id	name
1	Alice
2	Bob
3	Charlie
4	David
5	Emily

union all

id	name
1	Alice
2	Bob
3	Charlie
3	Charlie
4	David
5	Emily

intersect

id	name
3	Charlie

id	name
1	Alice ✓
2	Bob ✓

# Self Joins

20 February 2023 15:34

A self join is a type of join in which a table is joined with itself. This means that the table is treated as two separate tables, with each row in the table being compared to every other row in the same table.

Self joins are used when you want to compare the values of two different rows within the same table. For example, you might use a self join to compare the salaries of two employees who work in the same department, or to find all pairs of customers who have the same billing address.

user_id	name	age	emergency_contact
1	Nitish	34	11
2	Ankit	32	1
3	Neha	23	1
4	Radhika	34	3
8	Abhinav	31	11
11	Rahul	29	8

user_id	name	age	emergency_contact
1	Nitish	34	11
2	Ankit	32	1
3	Neha	23	1
4	Radhika	34	3
8	Abhinav	31	11
11	Rahul	29	8

name → emrg cont fr

Nitish	Rahul
Ankit	Nitish
Neha	Nitish

# Joining on more than one cols

20 February 2023 16:58

student

student_id	first_name	last_name	class_id	enrollment_year
1	John	Smith	1	2021
2	Jane	Doe	2	2020
3	Bob	Johnson	1	2021
4	Sally	Brown	3	2022
5	Tom	Williams	2	2022
6	Alice	Davis	4	2020

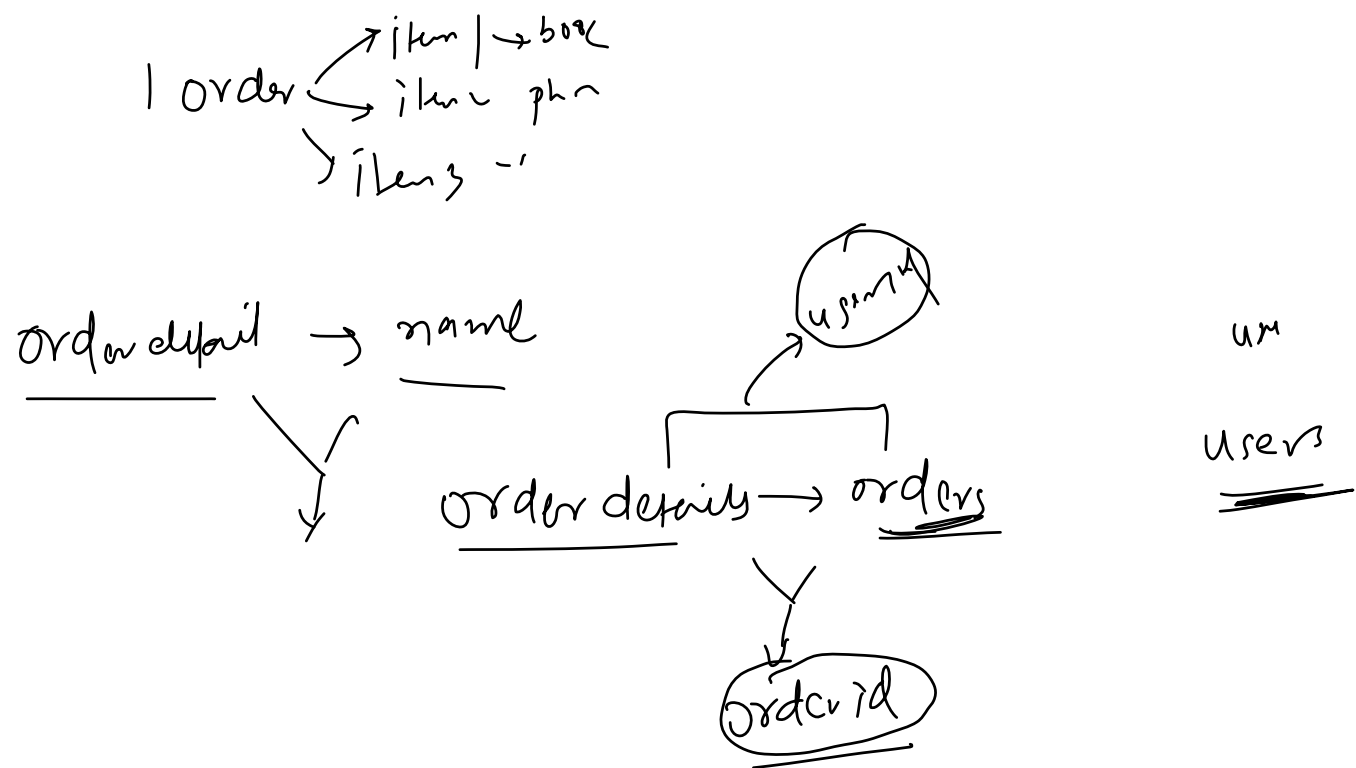
class - br

class_id	class_name	teacher	class_year
1	Math 101	Mr. Smith	2021
2	English 1	Ms. Johnson	2021
3	Science 1	Dr. Lee	2022
4	History 1	Ms. Williams	2022

# Joining more than 2 tables

20 February 2023 15:33

## 1. Find order name and corresponding category name



# Filtering Columns

20 February 2023 15:31

1. Find `order_id`, name and city by joining users and orders.
2. Find `order_id`, product category by joining `order_details` and `category`

# Filtering Rows

20 February 2023 15:32

1. Find all the orders placed in pune
2. Find all orders under Chairs category

## Practice Questions

20 February 2023 15:35

1. Find all profitable orders
2. Find the customer who has placed max number of orders
3. Which is the most profitable category
4. Which is the most profitable state
5. Find all categories with profit higher than 5000

