



SQL Joins (Inner, Left, Right and Full Join)

Last Updated : 17 Dec, 2024

SQL **joins** are the foundation of **database management systems**, enabling the combination of data from multiple tables based on relationships between columns. Joins allow **efficient data retrieval**, which is essential for generating meaningful observations and solving **complex business queries**.

Understanding SQL join types, such as **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN**, **FULL JOIN**, and **NATURAL JOIN**, is critical for working with relational databases.

In this article, we will cover the **different types of SQL joins**, including **INNER JOIN**, **LEFT OUTER JOIN**, **RIGHT JOIN**, **FULL JOIN**, and **NATURAL JOIN**. Each join type will be explained with examples, **syntax**, and practical use cases to help us understand when and how to use these joins effectively.

What is SQL Join?

SQL JOIN clause is used to **query** and **access data** from multiple tables by establishing **logical relationships** between them. It can access data from multiple tables simultaneously using common key values shared across different tables. We can use **SQL JOIN** with **multiple tables**. It can also be paired with other clauses, the most popular use will be using JOIN with **WHERE clause** to filter data retrieval.

Example of SQL JOINS

Consider the two tables, **Student** and **StudentCourse**, which share a common column **ROLL_NO**. Using SQL JOINS, we can combine data from these tables based on their **relationship**, allowing us to retrieve meaningful information like student details along with their **enrolled courses**.

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

StudentCourse Table

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

Both these tables are connected by one common key (column) i.e **ROLL_NO**. We can perform a JOIN operation using the given SQL query:

Query:

```
SELECT s.roll_no, s.name, s.address, s.phone, s.age, sc.course_id
FROM Student s
JOIN StudentCourse sc ON s.roll_no = sc.roll_no;
```

Output

ROLL_NO	NAME	ADDRESS	PHONE	AGE	COURSE_ID
---------	------	---------	-------	-----	-----------

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

1	HARSH	DELHI	XXXXXXXXXX	18	1
2	PRATIK	BIHAR	XXXXXXXXXX	19	2
3	RIYANKA	SILGURI	XXXXXXXXXX	20	2
4	DEEP	RAMNAGAR	XXXXXXXXXX	18	3
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19	1

Types of JOIN in SQL

There are many types of Joins in [SQL](#). Depending on the use case, we can use different type of **SQL JOIN** clause. Below, we explain the most commonly used join types with syntax and examples:

- [INNER JOIN](#)
- [LEFT JOIN](#)
- [RIGHT JOIN](#)
- [FULL JOIN](#)
- [Natural Join](#)

1. SQL INNER JOIN

The [INNER JOIN](#) keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the **result-set** by combining all rows from both the tables where the **condition satisfies** i.e value of the common field will be the same.

Syntax

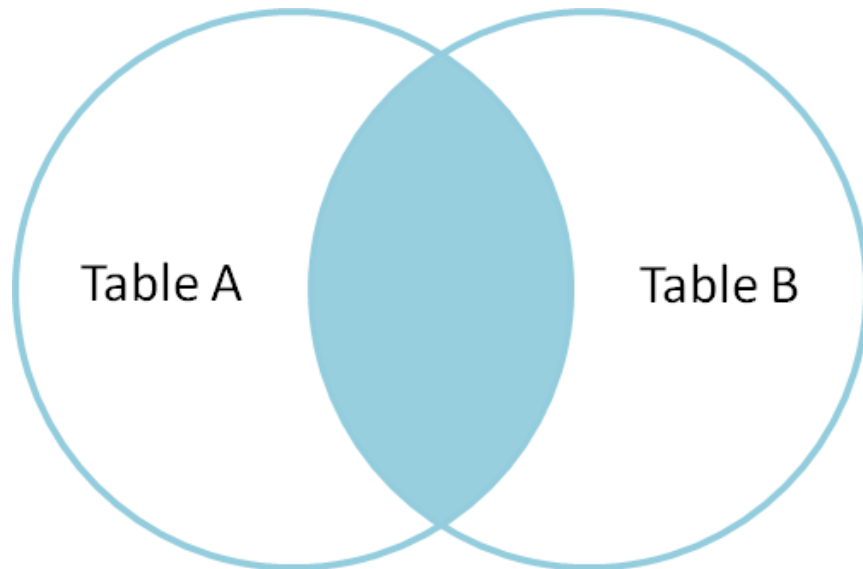
```
SELECT table1.column1,table1.column2,table2.column1,...
FROM table1
INNER JOIN table2
ON table1.matching_column = table2.matching_column;
```

Key Terms

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

- **table2**: Second table
- **matching_column**: Column common to both the tables.

Note: We can also write JOIN instead of INNER JOIN. JOIN is same as INNER JOIN.



INNER JOIN Example

Let's look at the example of **INNER JOIN** clause, and understand it's working. This query will show the names and age of students enrolled in different courses.

Query:

```
SELECT StudentCourse.COURSE_ID, Student.NAME, Student.AGE FROM Student  
INNER JOIN StudentCourse  
ON Student.ROLL_NO = StudentCourse.ROLL_NO;
```

Output

COURSE_ID	NAME	Age
1	HARSH	18
2	PRATIK	19
2	RIYANKA	20
3	DEEPA	19

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

2. SQL LEFT JOIN

LEFT JOIN returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is **no matching row** on the right side, the result-set will contain **null**. LEFT JOIN is also known as **LEFT OUTER JOIN**.

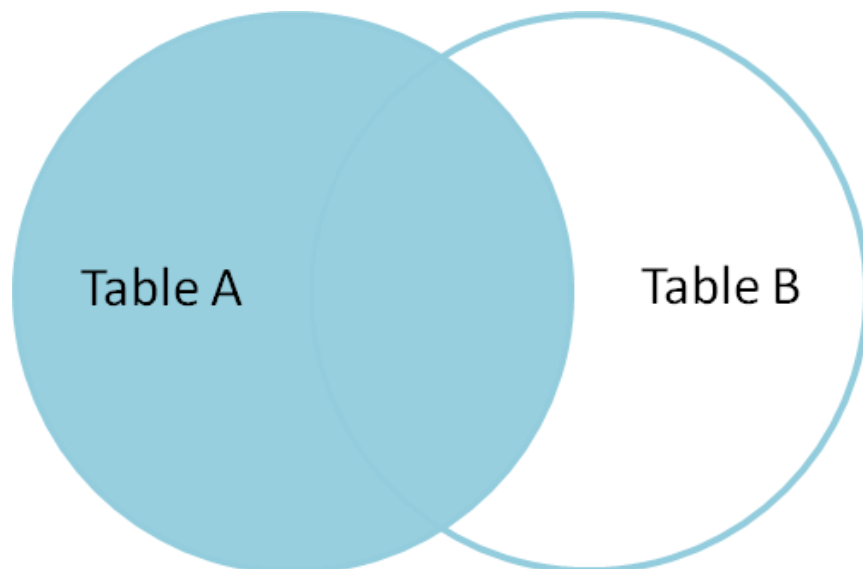
Syntax

```
SELECT table1.column1,table1.column2,table2.column1,....  
FROM table1  
LEFT JOIN table2  
ON table1.matching_column = table2.matching_column;
```

Key Terms

- **table1**: First table.
- **table2**: Second table
- **matching_column**: Column common to both the tables.

Note: We can also use LEFT OUTER JOIN instead of LEFT JOIN, both are the same.



LEFT JOIN Example

In this example, the **LEFT JOIN** retrieves all rows from the **Student** table and

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Query:

```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student
LEFT JOIN StudentCourse
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

Output

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL
ROHIT	NULL
NIRAJ	NULL

3. SQL RIGHT JOIN

RIGHT JOIN returns all the rows of the table on the **right side of the join** and matching rows for the table on the left side of the join. It is very similar to **LEFT JOIN** for the rows for which there is no matching row on the left side, the result-set will contain **null**. **RIGHT JOIN** is also known as **RIGHT OUTER JOIN**.

Syntax

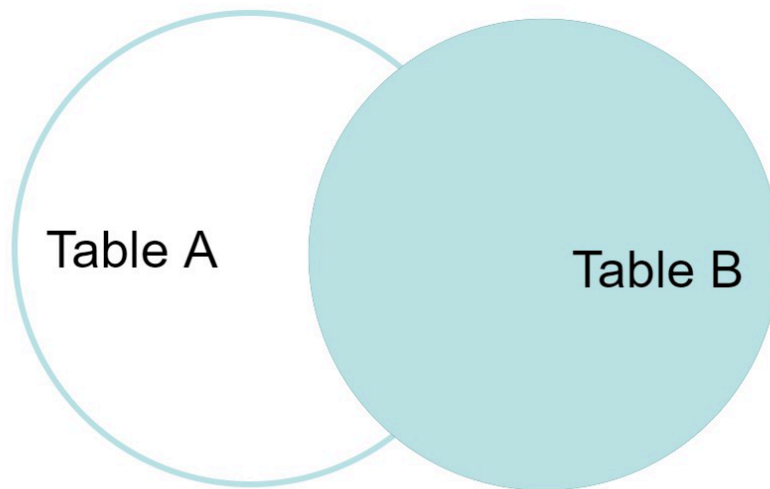
```
SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
RIGHT JOIN table2
ON table1.matching_column = table2.matching_column;
```

Key Terms

- **table1**: First table.
- **table2**: Second table

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Note: We can also use **RIGHT OUTER JOIN** instead of **RIGHT JOIN**, both are the same.



RIGHT JOIN Example

In this example, the **RIGHT JOIN** retrieves all rows from the **StudentCourse** table and the matching rows from the **Student** table based on the **ROLL_NO** column.

Query:

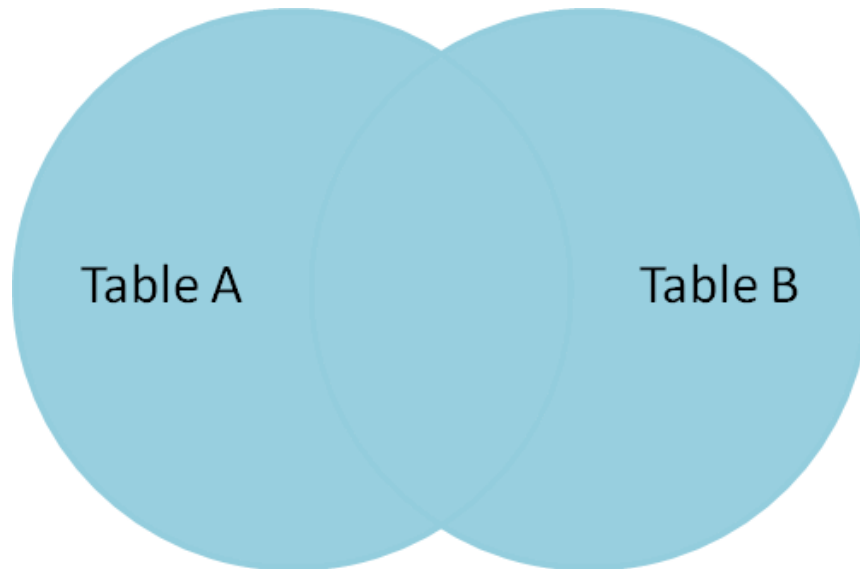
```
SELECT Student.NAME, StudentCourse.COURSE_ID  
FROM Student  
RIGHT JOIN StudentCourse  
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

Output

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
NULL	4
NULL	5
NULL	4

4. SQL FULL JOIN

FULL JOIN creates the result-set by combining results of both **LEFT JOIN** and **RIGHT JOIN**. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain *NULL* values.



Syntax

```
SELECT table1.column1,table1.column2,table2.column1,....  
FROM table1  
FULL JOIN table2  
ON table1.matching_column = table2.matching_column;
```

Key Terms

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

- **matching_column**: Column common to both the tables.

FULL JOIN Example

This example demonstrates the use of a **FULL JOIN**, which combines the results of both **LEFT JOIN** and **RIGHT JOIN**. The query retrieves all rows from the **Student** and **StudentCourse** tables. If a record in one table does not have a matching record in the other table, the result set will include that record with **NULL values** for the missing fields

Query:

```
SELECT Student.NAME, StudentCourse.COURSE_ID  
FROM Student  
FULL JOIN StudentCourse  
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

Output

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

NAME	COURSE_ID
NIRAJ	NULL
NULL	4
NULL	5
NULL	4

5. SQL Natural Join (?)

[Natural join](#) can join tables based on the **common columns** in the tables being joined. A natural join returns all rows by matching values in common columns having same name and **data type** of columns and that column should be present in both tables.

- Both table must have at least one common column with same column name and same data type.
- The two table are joined using **Cross join**.
- DBMS will look for a common column with same name and data type. Tuples having exactly same values in common columns are kept in result.

Natural join Example

Look at the two tables below- **Employee** and **Department**

Employee		
Emp_id	Emp_name	Dept_id
1	Ram	10

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Employee		
3	Bob	50

Department	
Dept_id	Dept_name
10	IT
30	HR
40	TIS

Problem: Find all Employees and their respective departments.

Solution Query: (Employee) ? (Department)

Emp_id	Emp_name	Dept_id	Dept_id	Dept_name
1	Ram	10	10	IT
2	Jon	30	30	HR
Employee data			Department data	

Conclusion

SQL joins are essential tools for anyone working with [relational databases](#). Understanding the different types of joins in **SQL**, like **INNER JOIN**, **LEFT OUTER JOIN**, **RIGHT JOIN**, and **FULL JOIN**, allows us to combine and query data effectively. With the **examples** and **syntax** covered here, we should feel confident applying these **SQL join types** to our data to retrieve **meaningful**

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

FAQs

What are the 4 types of join SQL?

In SQL, the four main types of joins are:

- *INNER JOIN*
- *LEFT JOIN*
- *RIGHT JOIN*
- *FULL JOIN*

What is a join in SQL?

A join in SQL is a relational database operation used to combine rows from two or more tables based on a related column between them.

What is the difference between INNER JOIN and LEFT JOIN?

The main difference between INNER JOIN and LEFT JOIN lies in how they handle unmatched rows. INNER JOIN focuses on matched rows only, while LEFT JOIN includes all rows from the left table, with NULLs where there is no match in the right table.

Joins in SQL

[Visit Course](#)[Comment](#)[More info](#)[Advertise with us](#)

Next Article

[SQL Inner Join](#)

Similar Reads

SQL for Data Science

Mastering SQL (Structured Query Language) has become a fundamental skill for anyone pursuing a career in data science. As data plays an increasingly central role in business and technology, SQL has emerged as the...

7 min read

Introduction to SQL

Setting Up the Environment

SQL Basics

SQL Operators

Working with Data

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Data Manipulation

SQL Joins (Inner, Left, Right and Full Join)

SQL joins are the foundation of database management systems, enabling the combination of data from multiple tables based on relationships between columns. Joins allow efficient data retrieval, which is...

6 min read

SQL Inner Join

SQL INNER JOIN is a powerful and frequently used operation in relational databases. It allows us to combine two or more tables based on a related column, returning only the records that satisfy the join condition. This...

4 min read

SQL Outer Join

SQL Outer Joins allow retrieval of rows from two or more tables based on a related column. Unlike inner joins, they also include rows that do not have a corresponding match in one or both of the tables. This...

4 min read

SQL Self Join

A Self Join in SQL is a powerful technique that allows one to join a table with itself. This operation is helpful when you need to compare rows within the same table based on specific conditions. A Self Join is often use...

4 min read

How to Group and Aggregate Data Using SQL?

In SQL, grouping and aggregating data are essential techniques for analyzing datasets. When dealing with large volumes of data, we often need to summarize or categorize it into meaningful groups. The combinatio...

4 min read

SQL HAVING Clause with Examples

The HAVING clause in SQL is used to filter query results based on aggregate functions. Unlike the WHERE clause, which filters individual rows before grouping, the HAVING clause filters groups of data after...

4 min read

Data Analysis

Data Visualization

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Corporate & Communications Address:**

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

About Us
Legal
Privacy Policy
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies**Python Tutorial**

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved