



(/)

<https://play.google.com/store/apps/details?id=com.studytonight.app>

SQL JOIN



SQL Join is used to fetch data from two or more tables, which is joined to appear as single set of data. It is used for combining column from two or more tables by using values common to both tables.

JOIN Keyword is used in SQL queries for joining two or more tables. Minimum required condition for joining table, is **(n-1)** where **n**, is number of tables. A table can also join to itself, which is known as, **Self Join**.



Types of JOIN

Following are the types of JOIN that we can use in SQL:



- Inner
- Outer
- Left
- Right



Cross JOIN or Cartesian Product

This type of JOIN returns the cartesian product of rows from the tables in Join. It will return a table which consists of records which combines each row from the first table with each row of the second table.

Cross JOIN Syntax is,

```
SELECT column-name-list
FROM
table-name1 CROSS JOIN table-name2;
```

Example of Cross JOIN

Following is the **class** table,

(<https://play.google.com/store/apps/details?id=com.studytonight.app>)

ID	NAME
1	abhi
2	adam
4	alex

and the **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Cross JOIN query will be,

```
SELECT * FROM
class CROSS JOIN class_info;
```

The resultset table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	1	DELHI
4	alex	1	DELHI
1	abhi	2	MUMBAI
2	adam	2	MUMBAI
4	alex	2	MUMBAI
1	abhi	3	CHENNAI

2	adam	3	CHENNAI	
4	alex	3	CHENNAI	



As you can see, this join returns the cross product of all the records present in both the tables.



INNER Join or EQUI Join

This is a simple JOIN in which the result is based on matched data as per the equality condition specified in the SQL query.

Inner Join Syntax is,

```
SELECT column-name-list FROM
table-name1 INNER JOIN table-name2
WHERE table-name1.column-name = table-name2.column-name;
```



Example of INNER JOIN

Consider a **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu

and the **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Inner JOIN query will be,

(<https://play.google.com/store/apps/details?id=com.studytonight.app>)

```
SELECT * from class INNER JOIN class_info where class.id = class_info.id;
```

The resultset table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI

Natural JOIN

Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.

The syntax for Natural Join is,

```
SELECT * FROM
table-name1 NATURAL JOIN table-name2;
```

Example of Natural JOIN

Here is the **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu

and the **class_info** table,

ID	Address
----	---------

1	DELHI	
2	MUMBAI	
3	CHENNAI	>

Natural join query will be,

```
SELECT * from class NATURAL JOIN class_info;
```

The resultset table will look like,

ID	NAME	Address
1	abhi	DELHI
2	adam	MUMBAI
3	alex	CHENNAI

In the above example, both the tables being joined have **ID** column(same name and same datatype), hence the records for which value of **ID** matches in both the tables will be the result of Natural Join of these two tables.

OUTER JOIN

Outer Join is based on both matched and unmatched data. Outer Joins subdivide further into,

1. Left Outer Join
2. Right Outer Join
3. Full Outer Join

LEFT Outer Join

The left outer join returns a resultset table with the **matched data** from the two tables and then the remaining rows of the **left** table and null from the **right** table's columns.

Syntax for Left Outer Join is,

SELECT column-name-list FROM
<https://play.google.com/store/apps/details?id=com.studytonight.app>
 ON table-name1.column-name = table-name2.column-name;

To specify a condition, we use the **ON** keyword with Outer Join.

Left outer Join Syntax for **Oracle** is,

SELECT column-name-list FROM
 table-name1, table-name2 on table-name1.column-name = table-name2.column-name(+);

Example of Left Outer Join

Here is the **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

and the **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Left Outer Join query will be,

```
SELECT * FROM class LEFT OUTER JOIN class_info ON (class.id = class_info.id);
```

(<https://play.google.com/store/apps/details?id=com.studytonight.app>)

The resultset table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
4	anu	null	null
5	ashish	null	null

RIGHT Outer Join

The right outer join returns a resultset table with the **matched data** from the two tables being joined, then the remaining rows of the **right** table and null for the remaining **left** table's columns.

Syntax for Right Outer Join is,

```
SELECT column-name-list FROM
table-name1 RIGHT OUTER JOIN table-name2
ON table-name1.column-name = table-name2.column-name;
```

Right outer Join Syntax for **Oracle** is,

```
SELECT column-name-list FROM
table-name1, table-name2
ON table-name1.column-name(+) = table-name2.column-name;
```

Example of Right Outer Join

Once again the **class** table,

ID	NAME
1	abhi

2	adam	
(https://play.google.com/store/apps/details?id=com.studytonight.app)		
3	alex	
4	anu	>
5	ashish	>

and the **class_info** table,

ID	Address	
1	DELHI	>
2	MUMBAI	>
3	CHENNAI	>
7	NOIDA	>
8	PANIPAT	>

Right Outer Join query will be,

```
SELECT * FROM class RIGHT OUTER JOIN class_info ON (class.id = class_info.id);
```

The resultant table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
null	null	7	NOIDA
null	null	8	PANIPAT

Full Outer Join

The full outer join returns a resultset table with the **matched data** of two table then remaining rows of both **left** table and then the **right** table.

(<https://play.google.com/store/apps/details?id=com.studytonight.app>)

Syntax of Full Outer Join is,

```
SELECT column-name-list FROM
table-name1 FULL OUTER JOIN table-name2
ON table-name1.column-name = table-name2.column-name;
```

Example of Full outer join is,

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

and the **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Full Outer Join query will be like,

```
SELECT * FROM class FULL OUTER JOIN class_info ON (class.id = class_info.id);
```

The resultset table will look like,

ID	NAME	ID	Address	
1	abhi	1	DELHI	(https://play.google.com/store/apps/details?id=com.studytonight.app)
2	adam	2	MUMBAI	
3	alex	3	CHENNAI	
4	anu	null	null	
5	ashish	null	null	
null	null	7	NOIDA	
null	null	8	PANIPAT	

[← Prev \(sql-function.php\)](#)[Next → \(sql-alias.php\)](#)[DBMS & SQL Tests](#)[\(/tests/?subject=dbms\)](/tests/?subject=dbms)

(<https://play.google.com/store/apps/details?id=com.studytonight.app>)

Boyce-Codd Normal Form (BCNF) for Beginners

(/post/bcnf-for-beginners)

Third Normal Form (3NF) Explained with help of a very Simple Example

(/post/third-normal-form-explained-with-very-simple-example)

The Simplest Explanation for Second Normal Form - Database Normalisation

(/post/the-simplest-explanation-for-second-normal-form-database-normalisation)

What is First Normal Form (1NF) of Database Normalisation?

(/post/what-is-first-normal-form-of-database-normalisation)

Basic Concept of Database Normalisation - Simple Explanation for Beginners

(/post/basic-concept-of-database-normalisation)

What is Studytonight?

About Us (/about)

Authors (/authors)

Collaborate (/collaborate)

Testimonials (/testimonials)

Privacy Policy (/privacy)

Terms (/terms)

Contact Us (/contact)

Suggest (/suggest)

Tutorials

Android (/android)

Core Java (/java)

<https://play.google.com/store/apps/details?id=com.studytonight.app>

Data Structures (/data-structures)

Python (/python)

Network Programming (/network-programming-in-python)

DBMS & SQL (/dbms)

Servlet (/servlet)

More... (/library)

Tests

Core Java (/tests)

Android (/tests/?subject=android)

C++ (/tests/?subject=cpp)

DBMS (/tests/?subject=dbms)

C Language (/tests/?subject=c)

More... (/tests)

Learn to Code

HTML (/code/html)

CSS (/cascading-style-sheet/)

Website Development (/code/playground)

Java Interview Question (/flashcards/Java)

C++ Interview Question (/flashcards/Cpp)

OS Interview Question (/flashcards/OS)

DBMS Interview Question (/flashcards/Sql)

More... (/flashcards)