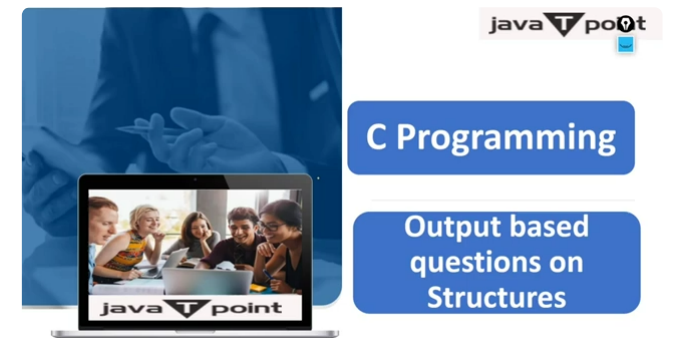


[Home](#) [DBMS](#) [SQL](#) [PL/SQL](#) [SQLite](#) [MongoDB](#) [Cassandra](#) [MySQL](#) [Oracle](#) [CouchDB](#) [Neo4j](#) [DB2](#) [C](#) [Java](#) [Projects](#)

↑ SCROLL TO TOP



# Join Operations:

A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied. It is denoted by  $\bowtie$ .

## Example:

### EMPLOYEE

EMP_CODE	EMP_NAME
101	Stephan
102	Jack
103	Harry

### SALARY

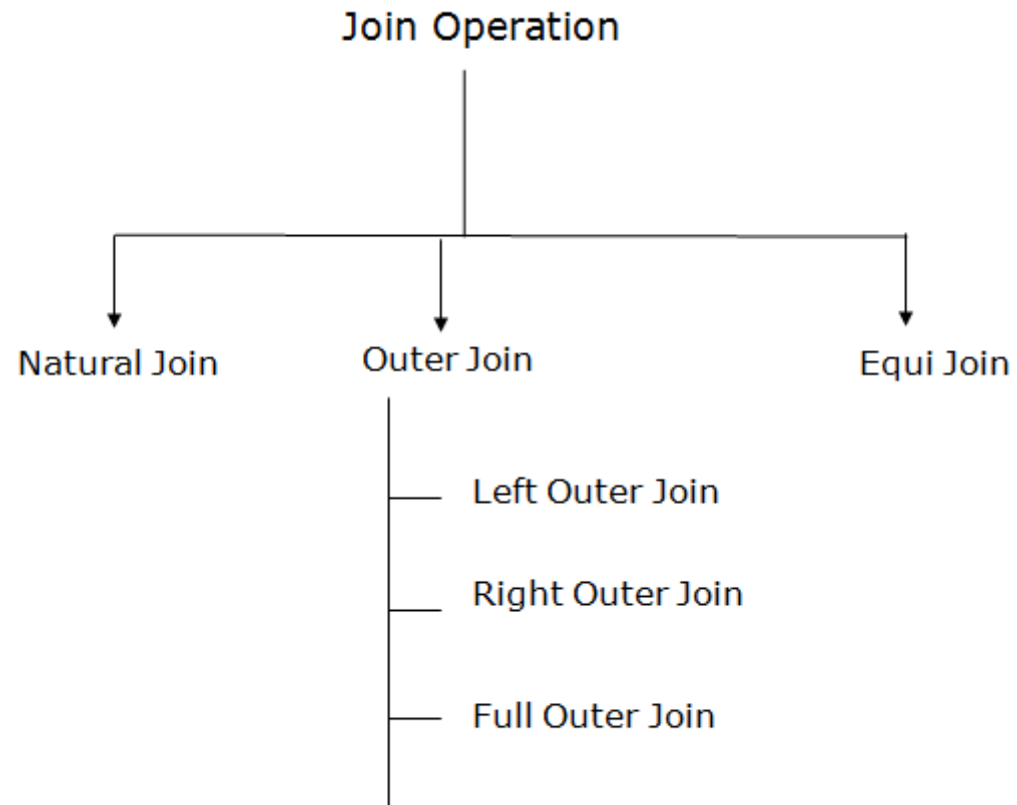
EMP_CODE	SALARY
101	50000
102	30000
103	25000

Operation: (EMPLOYEE  $\bowtie$  SALARY)

**Result:**

<b>EMP_CODE</b>	<b>EMP_NAME</b>	<b>SALARY</b>
101	Stephan	50000
102	Jack	30000
103	Harry	25000

## Types of Join operations:



### 1. Natural Join:

- A natural join is the set of tuples of all combinations in R and S that are equal on their common attribute names.
- It is denoted by  $\bowtie$ .

**Example:** Let's use the above EMPLOYEE table and SALARY table:

**Input:**

```
⋈ EMP_NAME, SALARY (EMPLOYEE ⋈ SALARY)
```

**Output:**

EMP_NAME	SALARY
Stephan	50000
Jack	30000
Harry	25000

## 2. Outer Join:

The outer join operation is an extension of the join operation. It is used to deal with missing information.

**Example:**

**EMPLOYEE**

EMP_NAME	STREET	CITY
Ram	Civil line	Mumbai

Shyam	Park street	Kolkata
Ravi	M.G. Street	Delhi
Hari	Nehru nagar	Hyderabad

**FACT\_WORKERS**

EMP_NAME	BRANCH	SALARY
Ram	Infosys	10000
Shyam	Wipro	20000
Kuber	HCL	30000
Hari	TCS	50000

**Input:**

(EMPLOYEE ⋈ FACT\_WORKERS)

**Output:**

EMP_NAME	STREET	CITY	BRANCH	SALARY
Ram	Civil line	Mumbai	Infosys	10000
Shyam	Park street	Kolkata	Wipro	20000
Hari	Nehru nagar	Hyderabad	TCS	50000

An outer join is basically of three types:

- Left outer join
- Right outer join
- Full outer join

### a. Left outer join:

- Left outer join contains the set of tuples of all combinations in R and S that are equal on their common attribute names.
- In the left outer join, tuples in R have no matching tuples in S.
- It is denoted by  $\bowtie$ .

**Example:** Using the above EMPLOYEE table and FACT\_WORKERS table

**Input:**

EMPLOYEE  $\bowtie$  FACT\_WORKERS

EMP_NAME	STREET	CITY	BRANCH	SALARY
----------	--------	------	--------	--------



Ram	Civil line	Mumbai	Infosys	10000
Shyam	Park street	Kolkata	Wipro	20000
Hari	Nehru street	Hyderabad	TCS	50000
Ravi	M.G. Street	Delhi	NULL	NULL

### b. Right outer join:

- Right outer join contains the set of tuples of all combinations in R and S that are equal on their common attribute names.
- In right outer join, tuples in S have no matching tuples in R.
- It is denoted by  $\bowtie_r$ .

**Example:** Using the above EMPLOYEE table and FACT\_WORKERS Relation

#### Input:

EMPLOYEE  $\bowtie_r$  FACT\_WORKERS

#### Output:

EMP_NAME	BRANCH	SALARY	STREET	CITY
Ram	Infosys	10000	Civil line	Mumbai
Shyam	Wipro	20000	Park street	Kolkata
Hari	TCS	50000	Nehru street	Hyderabad
Kuber	HCL	30000	NULL	NULL

### c. Full outer join:

- Full outer join is like a left or right join except that it contains all rows from both tables.
- In full outer join, tuples in R that have no matching tuples in S and tuples in S that have no matching tuples in R in their common attribute name.
- It is denoted by  $\bowtie$ .

**Example:** Using the above EMPLOYEE table and FACT\_WORKERS table

#### Input:

EMPLOYEE  $\bowtie$  FACT\_WORKERS

#### Output:

EMP_NAME	STREET	CITY	BRANCH	SALARY
Ram	Civil line	Mumbai	Infosys	10000
Shyam	Park street	Kolkata	Wipro	20000
Hari	Nehru street	Hyderabad	TCS	50000
Ravi	M.G. Street	Delhi	NULL	NULL
Kuber	NULL	NULL	HCL	30000

### 3. Equi join:

It is also known as an inner join. It is the most common join. It is based on matched data as per the equality condition. The equi join uses the comparison operator(=).

#### Example:

#### CUSTOMER RELATION

CLASS_ID	NAME
1	John
2	Harry
3	Jackson

**PRODUCT**

PRODUCT_ID	CITY
1	Delhi
2	Mumbai
3	Noida

**Input:**

CUSTOMER ⋈ PRODUCT
--------------------

**Output:**

CLASS_ID	NAME	PRODUCT_ID	CITY
1	John	1	Delhi
2	Harry	2	Mumbai
3	Harry	3	Noida

[← Prev](#)[Next →](#)

For Videos Join Our Youtube Channel: [Join Now](#)

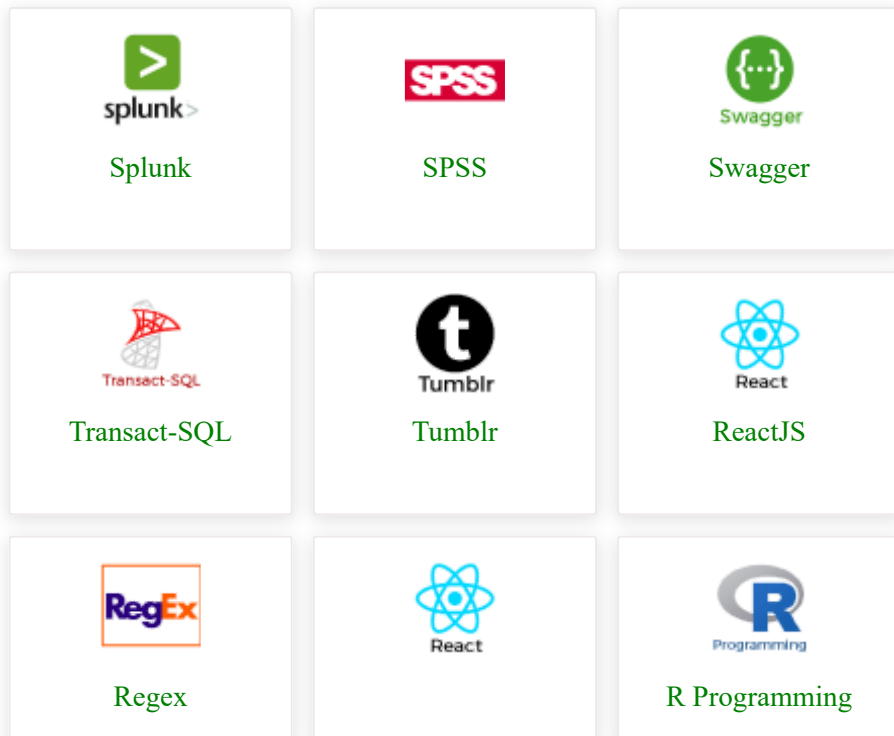
## Feedback

- Send your Feedback to [feedback@javatpoint.com](mailto:feedback@javatpoint.com)

## Help Others, Please Share



## Learn Latest Tutorials



## Reinforcement



RxJS

## Learning



React Native

Python Design  
PatternsPython Pillow  
tutorial

Python Pillow

Python Turtle  
tutorial













Python Turtle

Keras tutorial  
Keras

## Preparation

Aptitude  
AptitudeLogical  
Reasoning  
ReasoningVerbal Ability  
Verbal AbilityInterview  
Questions  
Interview QuestionsCompany  
Interview  
Questions  
Company Questions


## Trending Technologies


 Artificial Intelligence Tutorial  Artificial Intelligence	 AWS Tutorial  AWS	 Selenium tutorial  Selenium
 Cloud Computing tutorial  Cloud Computing	 Hadoop tutorial  Hadoop	 ReactJS Tutorial  ReactJS
 Data Science Tutorial  Data Science	 Angular 7 Tutorial  Angular 7	 Blockchain Tutorial  Blockchain
 Git Tutorial  Git	 Machine Learning Tutorial  Machine Learning	 DevOps Tutorial  DevOps





## B.Tech / MCA

 DBMS tutorial  
DBMS


 Data Structures  
tutorial  
Data Structures


 DAA tutorial  
DAA

 Operating  
System tutorial  
Operating System

 Computer  
Network tutorial  
Computer Network

 Compiler  
Design tutorial  
Compiler Design

 Computer  
Organization and  
Architecture  
Computer  
Organization

 Discrete  
Mathematics  
Tutorial  
Discrete  
Mathematics

 Ethical Hacking  
Tutorial  
Ethical Hacking



