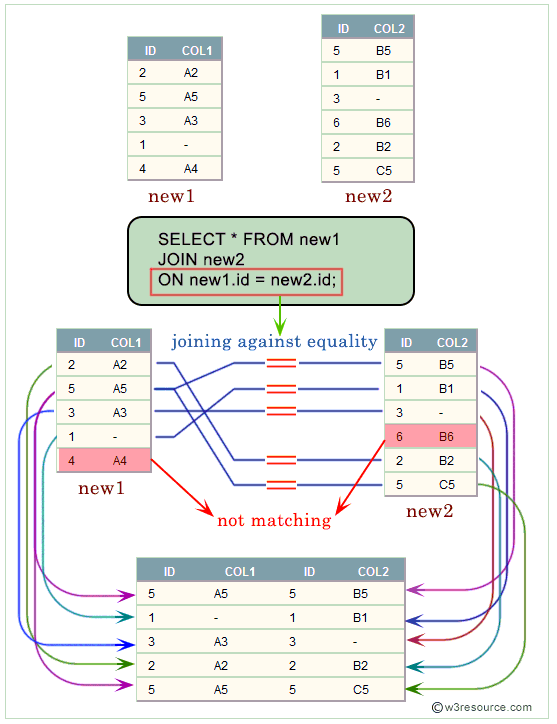
SQL JOIN

EQUI JOIN

SQL EQUI JOIN performs a JOIN against equality or matching column(s) values of the associated tables. An equal sign (=) is used as comparison operator in the where clause to refer equality.

You may also perform EQUI JOIN by using JOIN keyword followed by ON keyword and then specifying names of the columns along with their associated tables to check equality.



1. Inner join can have equality (=) and other operators (like <,>,<>) in the join condition.
2. Equi join only have equality (=) operator in the join condition.

Table name — Student

| id | name | class | city |
| --- | --- | --- | --- |
| 3 | Hina | 3 | Delhi |
| 4 | Megha | 2 | Delhi |
| 6 | Gouri | 2 | Delhi |

Table name — Record

| id | class | city |
| --- | --- | --- |
| 9 | 3 | Delhi |
| 10 | 2 | Delhi |
| 12 | 2 | Delhi |

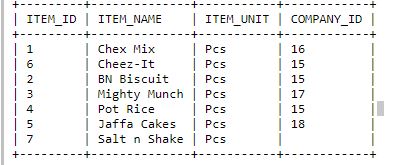
SELECT student.name, student.id, record.class, record.city FROM student, record WHERE student.city = record.city;

SELECT student.name, student.id, record.class, record.city FROM student JOIN record ON student.city = record.city;

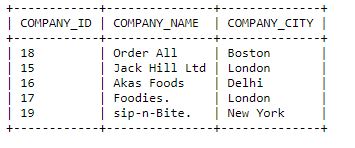
| name | id | class | city |
| --- | --- | --- | --- |
| Hina | 3 | 3 | Delhi |
| Megha | 4 | 3 | Delhi |
| Gouri | 6 | 3 | Delhi |
| Hina | 3 | 2 | Delhi |
| Megha | 4 | 2 | Delhi |
| Gouri | 6 | 2 | Delhi |
| Hina | 3 | 2 | Delhi |
| Megha | 4 | 2 | Delhi |
| Gouri | 6 | 2 | Delhi |

**Natural Join**

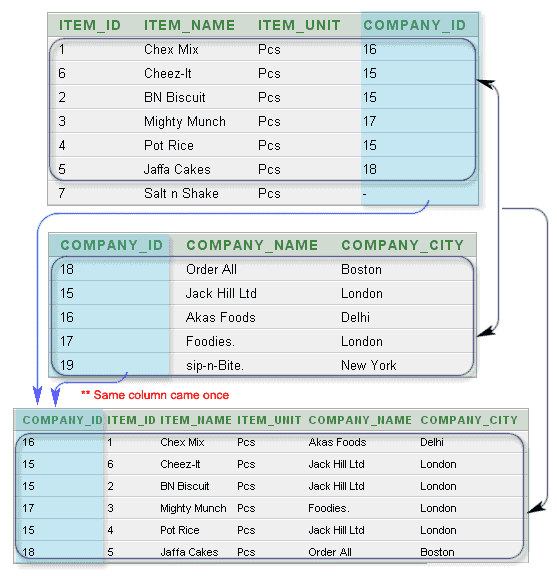
**Sample table: foods**



Sample table



SELECT \* FROM foods NATURAL JOIN company;



Inner join

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.’

OrderID CustomerID OrderDate

10308 2 1996-09-18

10309 37 1996-09-19

10310 77 1996-09-20

CustomerID CustomerName ContactName C ountry

1 Alfreds Futterkiste Maria Anders Germany

2 Ana Trujillo Emparedados Ana Trujillo Mexico

3 Antonio Moreno Taquería Antonio Moreno Mexico

# SQL INNER JOIN Keyword



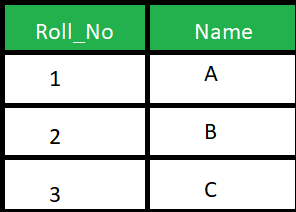
SELECT Orders.OrderID, Customers.CustomerName FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

## JOIN Three Tables

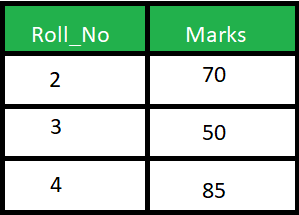
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName  
FROM ((OrdersINNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID) INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

Difference between Natural join and Inner join

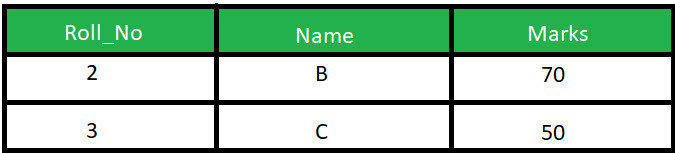
Student table



Marks table

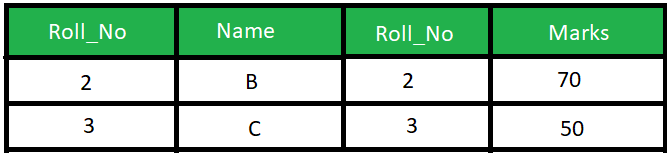


SELECT \* FROM Student S NATURAL JOIN Marks M;



**Inner Join :**

SELECT \* FROM student S INNER JOIN Marks M ON S.Roll\_No = M.Roll\_No;



|  |  |  |
| --- | --- | --- |
| **Natural Join** | **Equi Join** | **Inner Join** |
| It joins the tables based on the same column names and their data types. | It joins the tables based on the equality or matching column values in the associated tables. | It joins the tables based on the column name specified in the ON clause explicitly. It returns only those rows that exist in both tables. |
| It always returns unique columns in the result set. | It can return all attributes of both tables along with duplicate columns that match the join condition. | It returns all the attributes of both tables along with duplicate columns that match the ON clause condition. |
| The syntax of a natural join is given below:  SELECT [column\_names | \*] FROM table\_name1 NATURAL JOIN table\_name2; | The syntax of equijoin is given below: SELECT column\_name (s)  FROM table\_name1, table\_name2, ...., table\_nameN WHERE table\_name1.column\_name = table\_name2.column\_name; | The syntax of inner join is given below: SELECT [column\_names | \*]  FROM table\_name1  INNER JOIN table\_name2 ON table\_name1.column\_name = table\_name2.column\_name; |

# SQL LEFT JOIN Keyword

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;

## SQL RIGHT JOIN Keyword

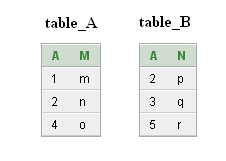
## The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID ORDER BY Orders.OrderID;

# SQL FULL OUTER JOIN Keyword

## SQL FULL OUTER JOIN Keyword

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.



**SELECT \* FROM table\_A FULL OUTER JOIN table\_B ON table\_A.A=table\_B.A;**

Output:



**select \* from student left join mark on student.studentid = mark.studentid union all select \* from student right join mark on student.studentid = mark.studentid;**

SELF JOIN

A self join is a join in which a table is joined with itself (which is also called Unary relationships), especially when the table has a FOREIGN KEY which references its own PRIMARY KEY. To join a table itself means that each row of the table is combined with itself and with every other row of the table.

CREATE TABLE employee(emp\_id varchar(5) NOT NULL,

emp\_name varchar(20) NULL,

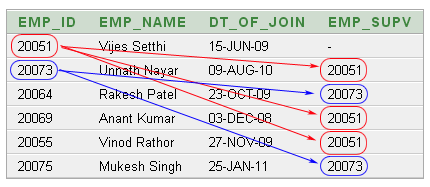
dt\_of\_join date NULL,

emp\_supv varchar(5) NULL,

CONSTRAINT emp\_id PRIMARY KEY(emp\_id) ,

CONSTRAINT emp\_supv FOREIGN KEY(emp\_supv)

REFERENCES employee(emp\_id));



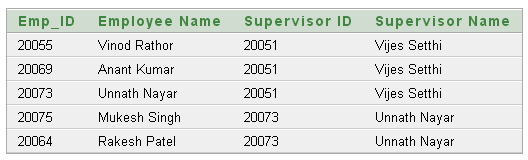
list of employees and their supervisor the following SQL statement has used:

SELECT a.emp\_id AS "Emp\_ID",a.emp\_name AS "Employee Name",

b.emp\_id AS "Supervisor ID",b.emp\_name AS "Supervisor Name"

FROM employee a, employee b

WHERE a.emp\_supv = b.emp\_id;

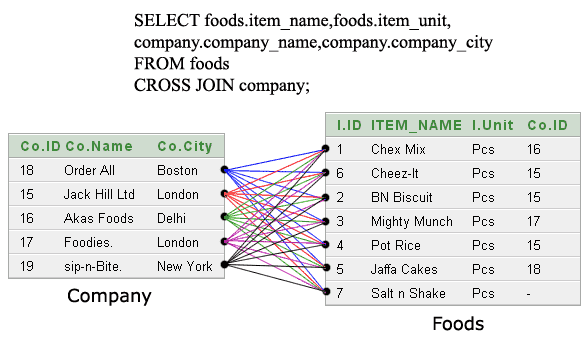


What is a CROSS JOIN?

Cross joins are used to return every combination of rows from two tables, this sometimes called a Cartesian product.

CROSS JOIN to join two or more unrelated tables.





## Example

## Perform cross join for this example.

