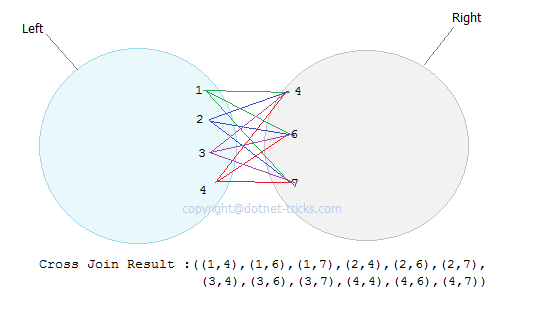
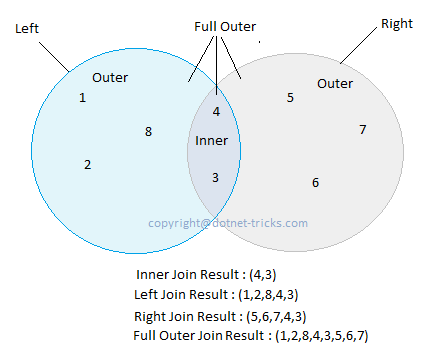
Types of Joins

In Sql Server we have only three types of joins. Using these joins we fetch the data from multiple tables based on condition.

1. Inner Join

Inner join returns only those records/rows that match/exists in both the tables. Syntax for Inner Join is as

* 1. **Select \* from table\_1 as t1**
  2. **inner join table\_2 as t2**
  3. **on t1.IDcol=t2.IDcol**



1. Outer Join

We have three types of Outer Join.

* 1. Left Outer Join

Left outer join returns all records/rows from left table and from right table returns only matched records. If there are no columns matching in the right table, it returns NULL values. Syntax for Left outer Join is as :

* + 1. **Select \* from table\_1 as t1**
    2. **left outer join table\_2 as t2**
    3. **on t1.IDcol=t2.IDcol**
  1. Right Outer Join

Right outer join returns all records/rows from right table and from left table returns only matched records. If there are no columns matching in the left table, it returns NULL values. Syntax for right outer Join is as :

* + 1. **Select \* from table\_1 as t1**
    2. **right outer join table\_2 as t2**
    3. **on t1.IDcol=t2.IDcol**
  1. Full Outer Join

Full outer join combines left outer join and right outer join. This join returns all records/rows from both the tables.If there are no columns matching in the both tables, it returns NULL values. Syntax for full outer Join is as :

* + 1. **Select \* from table\_1 as t1**
    2. **full outer join table\_2 as t2**
    3. **on t1.IDcol=t2.IDcol**

1. Cross Join

Cross join is a cartesian join means cartesian product of both the tables. This join does not need any condition to join two tables. This join returns records/rows that are multiplication of record number from both the tables means each row on left table will related to each row of right table. Syntax for right outer Join is as :

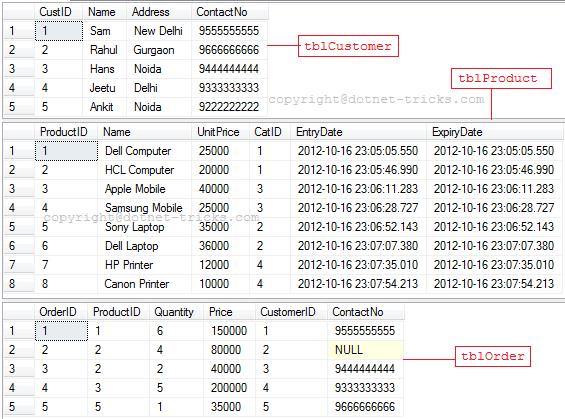
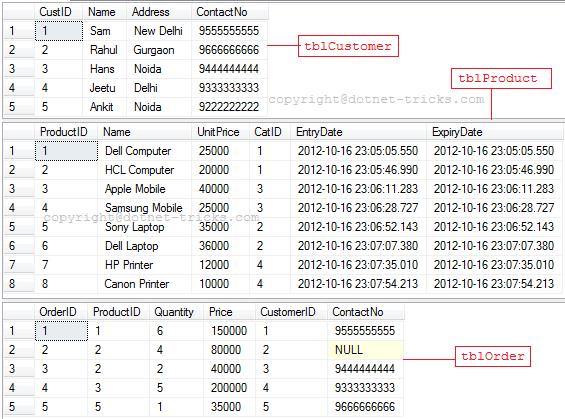
* 1. **Select \* from table\_1**
  2. **cross join table\_2**

1. Self Join

Self join is used to join a database table to itself, particularly when the table has a Foreign key that references its own Primary Key. Basically we have only three types of joins : Inner join, Outer join and Cross join. We use any of these three JOINS to join a table to itself. Hence Self join is not a type of Sql join.

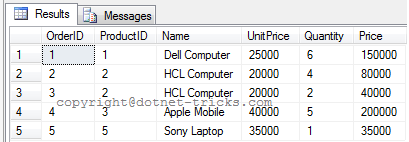
Join Examples

Suppose we following three tables and data in these three tables is shown in figure. You can download the SQL script used in this article by using [link](https://www.box.com/s/ynljdutllx0xn9y85x9p).



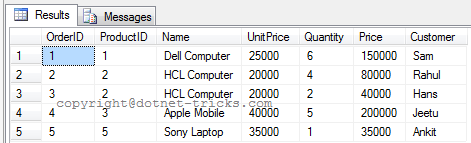
Inner Join

1. **SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price**
2. **FROM tblProduct AS t0**
3. **INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **ORDER BY t1.OrderID**



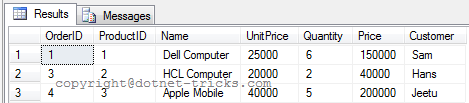
Inner Join among more than two tables

1. **SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name AS Customer**
2. **FROM tblProduct AS t0**
3. **INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID**
5. **ORDER BY t1.OrderID**



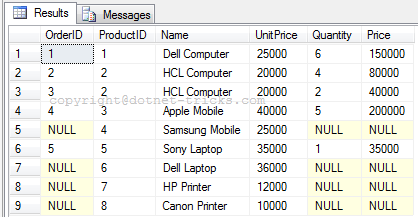
Inner Join on multiple conditions

1. **SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name AS Customer**
2. **FROM tblProduct AS t0**
3. **INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID AND t1.ContactNo = t2.ContactNo**
5. **ORDER BY t1.OrderID**



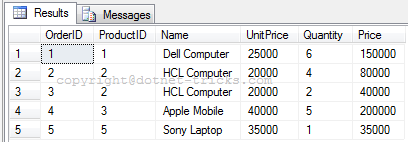
Left Outer Join

1. **SELECT t1.OrderID AS OrderID , t0.ProductID , t0.Name , t0.UnitPrice , t1.Quantity AS Quantity , t1.Price AS Price**
2. **FROM tblProduct AS t0**
3. **LEFT OUTER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **ORDER BY t0.ProductID**



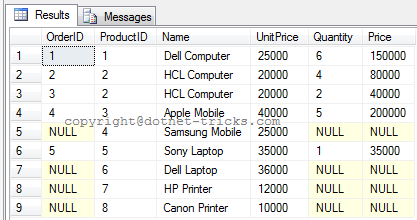
Right Outer Join

1. **SELECT t1.OrderID AS OrderID , t0.ProductID , t0.Name , t0.UnitPrice , t1.Quantity AS Quantity , t1.Price AS Price**
2. **FROM tblProduct AS t0**
3. **RIGHT OUTER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **ORDER BY t0.ProductID**



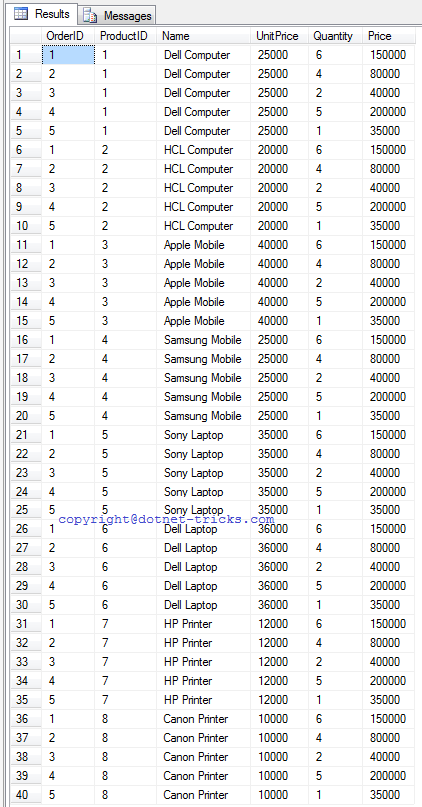
Full Outer Join

1. **SELECT t1.OrderID AS OrderID , t0.ProductID , t0.Name , t0.UnitPrice , t1.Quantity AS Quantity , t1.Price AS Price**
2. **FROM tblProduct AS t0**
3. **FULL OUTER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID**
4. **ORDER BY t0.ProductID**



Cross Join

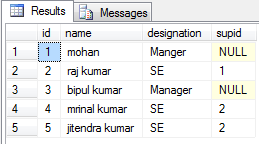
1. **SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price**
2. **FROM tblProduct AS t0, tblOrder AS t1**
3. **ORDER BY t0.ProductID**



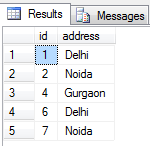
Self Join

To understand Self Join, suppose we following two tables and data in these two tables is shown in figure.

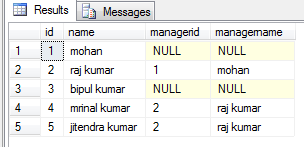
1. **CREATE TABLE emp**
2. **(**
3. **id int NOT NULL primary key,**
4. **name varchar(100) NULL,**
5. **designation varchar(50) NULL,**
6. **supid int foreign key references emp(id) ) *-- In this table we have a Foreign key supid that references its own Primary Key id. We use it for Self Join***
7. **INSERT INTO emp(id,name,designation) VALUES(1,'mohan','Manger')**
8. **INSERT INTO emp(id,name,designation,supid) VALUES(2,'raj kumar','SE',1)**
9. **INSERT INTO emp(id,name,designation) VALUES(3,'bipul kumar','Manager')**
10. **INSERT INTO emp(id,name,designation,supid) VALUES(4,'mrinal kumar','SE',2)**
11. **INSERT INTO emp(id,name,designation,supid) VALUES(5,'jitendra kumar','SE',2)**



1. **CREATE TABLE empinfo**
2. **(**
3. **id int primary key,**
4. **address varchar(50) NULL**
5. **)**
6. **INSERT INTO empinfo(id,address) VALUES(1,'Delhi')**
7. **INSERT INTO empinfo(id,address) VALUES(2,'Noida')**
8. **INSERT INTO empinfo(id,address) VALUES(4,'Gurgaon')**
9. **INSERT INTO empinfo(id,address) VALUES(6,'Delhi')**
10. **INSERT INTO empinfo(id,address) VALUES(7,'Noida')**



1. **select e.id,e.name,e.supid as managerid, ei.name as managername from emp e left join emp ei on e.supid=ei.id;**
2. ***-- outer keyword is optional***



Difference between inner join and equi join and natural join

SQL join clause is used to retrieve data from two or more database tables. In previous article, I have explained the [Different Types of SQL Joins](http://www.dotnet-tricks.com/Tutorial/sqlserver/W1aI140312-Different-Types-of-SQL-Joins.html). In this article, I would explain the difference among inner join, equi join and natural join.

Inner Join

This is the most used join in the SQL. this join returns only those records/rows that match/exists in both the database tables.

Inner Join Example

1. **SELECT \* FROM tblEmp JOIN tblDept**
2. **ON tblEmp.DeptID = tblDept.DeptID;**

In the join condition, you can also use other operators like <,>,<>.

Equi Join

Equi join is a special type of join in which we use only equality operator. Hence, when you make a query for join using equality operator then that join query comes under Equi join.

Equi Join Example

1. **SELECT \* FROM tblEmp JOIN tblDept**
2. **ON tblEmp.DeptID = tblDept.DeptID;**
3. ***--Using Clause is not supported by SQL Server***
4. ***--Oracle and MySQL Query***
5. **SELECT \* FROM tblEmp INNER JOIN tblDept USING(DeptID)**

Note

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.
3. Equi join can be an Inner join, Left Outer join, Right Outer join
4. The USING clause is not supported by SQL Server and Sybase. This clause is supported by Oracle and MySQL.

Natural Join

Natural join is a type of equi join which occurs implicitly by comparing all the same names columns in both tables. The join result have only one column for each pair of equally named columns.

Natural Join Example

1. ***--Run in Oracle and MySQL***
2. **SELECT \* FROM tblEmp NATURAL JOIN tblDept**

Note

1. In Natural join, you can't see what columns from both the tables will be used in the join. In Natural join, you might not get the desired result what you are expecting.
2. Natural join clause is not supported by SQL Server, it is supported by Oracle and MySQL.