**Different Types of SQL Join:**

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

**Different Types Of SQL Join:**

**Inner Join:** Returns records that have matching values on both ends.

**Left(Outer) Join:** Returns all the records from the left table and the matched record from the right table.

**Right(Outer) Join:**

Returns all the records from the right table and the matched record from the left table

**Full(Outer) Join:**

Returns all the records from left and right table where there is a match in either left or right table.

**Inner Join:**

The first table is created like this:

create table Orders(OrderID INT NOT NULL, CustomerID INT NOT NULL,EmployeeID INT NOT NULL, OrderDate DATE, ShipperID INT NOT NULL, PRIMARY KEY(OrderID));

And the second table is created like this:

create table Customers(CustomerID INT NOT NULL AUTO\_INCREMENT, CustomerName VARCHAR(80) NOT NULL, ContactName VARCHAR(80) NOT NULL, Address VARCHAR(80) NOT NULL, City VARCHAR(20) NOT NULL, PosalCode INT NOT NULL,Country VARCHAR(20) NOT NULL, PRIMARY KEY(customerID));

Now, let’s insert some data in them:

Suppose, the following insertions took place:

insert into Orders(OrderID, CustomerID, EmployeeID, OrderDate, ShipperID) VALUES(1,1,1,’2017-05-02’,1);

insert into Orders(OrderID, CustomerID, EmployeeID, OrderDate, ShipperID) VALUES(2,1,1,’2017-05-03’,1);

insert into Orders(OrderID, CustomerID, EmployeeID, OrderDate, ShipperID) VALUES(3,2,2,’2017-05-04’,2);

And, the table looks like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **EmployeeID** | **OrderDate** | **ShipperID** |
| 1 | 1 | 1 | 2017-05-02 | 1 |
| 2 | 1 | 1 | 2017-05-03 | 1 |
| 3 | 2 | 2 | 2017-05-03 | 2 |

Now, after that,

Let’s insert some data in table Customers.

After inserting some data in Customers table, it looks like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer**  **ID** | **CustomerName** | **Contact**  **Name** | **Address** | **City** | **Country** | **Postal**  **Code** |
| 1 | Sayak Haldar | Sayak Haldar | j-2,102, DDA Flats, Kalkaji | New Delhi | India | 110019 |
| 2 | Sayantan Pandit | Sayantan Pandit | Andul Purbopara, Howrah | Howrah | India | 711302 |
| 3 | Suman Banerjee | Suman Banerjee | Andul Mouri, Howrah | Howrah | India | 711302 |

Now, let’s see, a inner join query:

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

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **OrderDate** |
| 1 | Sayak Haldar | 2017-05-02 |
| 2 | Sayak Haldar | 2017-05-02 |
| 3 | Sayantan Pandit | 2017-05-03 |

We have have inner join on more than 2 tables:

Now, suppose, alongside with Orders and Customers their exist a third table name **ShipperInformation**:

**The table created with the following create table statement:**

create table ShipperInformation(ID INT NOT NULL AUTO\_INCREMENT, NAME VARCHAR(80) NOT NULL, PRIMARY KEY(ID));

And the dataset looks like:

|  |  |
| --- | --- |
| **ID** | **Name** |
| 1 | Amalendu Mallick |
| 2 | Debabrata Haldar |

SELECT Orders.OrderID, Customers.CustomerName, ShipperInformation.Name FROM ((Orders INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID) INNER JOIN ShipperInformation ON Orders.ShipperID = ShipperInformation.ID);

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **Name** |
| 1 | Sayak Haldar | Amalendu Mallick |
| 2 | Sayak Haldar | Amalendu Mallick |
| 3 | Sayantan Pandit | Debabrata Haldar |

**Left Join:**

Now, the Customers table looks like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer**  **ID** | **CustomerName** | **Contact**  **Name** | **Address** | **City** | **Country** | **Postal**  **Code** |
| 1 | Sayak Haldar | Sayak Haldar | j-2,102, DDA Flats, Kalkaji | New Delhi | India | 110019 |
| 2 | Sayantan Pandit | Sayantan Pandit | Andul Purbopara, Howrah | Howrah | India | 711302 |
| 3 | Suman Banerjee | Suman Banerjee | Andul Mouri, Howrah | Howrah | India | 711302 |

And the Orders table looks like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **EmployeeID** | **OrderDate** | **ShipperID** |
| 1 | 1 | 1 | 2017-05-02 | 1 |
| 2 | 1 | 1 | 2017-05-03 | 1 |
| 3 | 2 | 2 | 2017-05-03 | 2 |

**Now, the left join query is performed:**

Select Orders.OrderID, Customers.CustomerName, Orders.OrderDate from Orders LEFT JOIN Customers On Orders.CustomerID=Customers.CustomerID;

Or,

Select Orders.OrderID, Customers.CustomerName, Orders.OrderDate from Orders LEFT INNER JOIN Customers On Orders.CustomerID=Customers.CustomerID;

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **OrderDate** |
| 1 | Sayak Haldar | 2017-05-02 |
| 2 | Sayak Haldar | 2017-05-02 |
| 3 | Sayantan Pandit | 2017-05-03 |

Now, it can be seen, that both left join and inner join results same. Because, the left table (here, Orders) does not contain any data which is not related to Customers table.

Now, enter some data into the Orders table which is not related to Customers table:

insert INTO Orders(OrderID, CustomerID, EmployeeID , OrderDate , ShipperID) Values(4, 4, 4, '2017-05-04',3);

Now, the table Orders looks like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **EmployeeID** | **OrderDate** | **ShipperID** |
| 1 | 1 | 1 | 2017-05-02 | 1 |
| 2 | 1 | 1 | 2017-05-03 | 1 |
| 3 | 2 | 2 | 2017-05-03 | 2 |
| 4 | 4 | 4 | 2017-05-04 | 3 |

Now, perform the same join query:

Now, the result becomes

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **OrderDate** |
| 1 | Sayak Haldar | 2017-05-02 |
| 2 | Sayak Haldar | 2017-05-02 |
| 3 | Sayantan Pandit | 2017-05-03 |
| 4 | NULL | 2017-05-04 |

Now, the difference can be seen.

**Note:** The LEFT JOIN keyword returns all records from the left table (Customers), even if there are no matches in the right table (Orders).

Right Join:

Suppose, the Orders table looks like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **EmployeeID** | **OrderDate** | **ShipperID** |
| 1 | 1 | 1 | 2017-05-02 | 1 |
| 2 | 1 | 1 | 2017-05-03 | 1 |
| 3 | 2 | 2 | 2017-05-03 | 2 |
| 4 | 4 | 4 | 2017-05-04 | 3 |

**And, the Customers table looks like:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer**  **ID** | **CustomerName** | **Contact**  **Name** | **Address** | **City** | **Country** | **Postal**  **Code** |
| 1 | Sayak Haldar | Sayak Haldar | j-2,102, DDA Flats, Kalkaji | New Delhi | India | 110019 |
| 2 | Sayantan Pandit | Sayantan Pandit | Andul Purbopara, Howrah | Howrah | India | 711302 |
| 3 | Suman Banerjee | Suman Banerjee | Andul Mouri, Howrah | Howrah | India | 711302 |

Now, the right join query:

**Select Orders.OrderID, Customers.CustomerName, Orders.OrderDate from Orders RIGHT JOIN Customers On Orders.CustomerID=Customers.CustomerID;**

**Or,**

**Select Orders.OrderID, Customers.CustomerName, Orders.OrderDate from Orders RIGHT OUTER JOIN Customers On Orders.CustomerID=Customers.CustomerID;**

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **OrderDate** |
| 1 | Sayak Haldar | 2017-05-02 |
| 2 | Sayak Haldar | 2017-05-02 |
| 3 | Sayantan Pandit | 2017-05-03 |
| NULL | Suman Banerjee | NULL |

Now, right join  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.

**Full Outer Join:**

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

**Note:** FULL OUTER JOIN can potentially return very large result-sets!

However, FULL OUTER JOIN cannot be performed (I guess) in several/high versions of Mysql. Actually, we cannot perform FULL joins (Full Outer Join) in Mysql. (version independently)

Self Join:

Now, there’s a great example of self join:

To display all employees who earn less than their managers:

**Select E.ENAME EMP\_NAME, E.SAL EMP\_SAL, M.SAL FROM EMP E, EMP M Where E.MGRNO=M.ENO and E.SAL< M.SAL;**