Introtallent

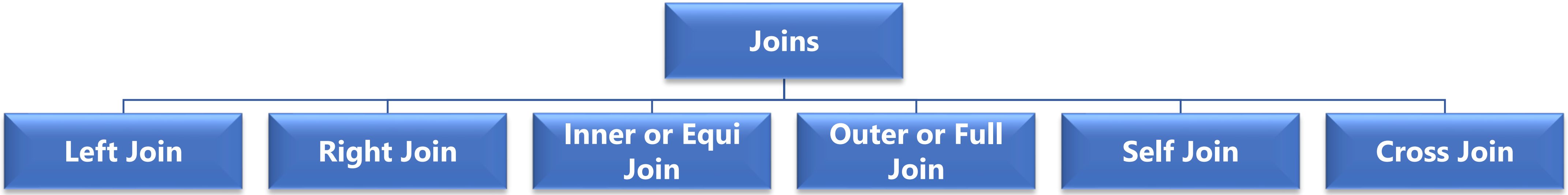
Training | Analytics | Consulting

Helping People Decode Analytics for Business

JOINS

Joins

In SQL, we use join when we need to extract data from more than one tables. In that case we can join both the table using the primary key columns and extract data as per the need. Different types of joins are:



Let’s assume that we have two tables in the training database (as represented below):

Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim

Fin

Id	Salary
2	200
3	300
4	400

Now, we need to combine Emp_id, Name and Salary together. In order to achieve that, we can use one of the joins depending upon how we want to combine the data. The common key (Primary Key) in both the table is Emp_id and Id. We will be using these keys to establish relationship while writing the query based on these two tables.

Let’s create the required tables first

```
/*Create Emp table*/
CREATE table training.emp(Emp_id int Primary Key,Name varchar(20));

/*Insert values into Emp table*/
INSERT into training.emp(Emp_id,Name) values(1,'Jim'),(2,'Kim'),(3,'Lim’);

/*Create Fin table*/
CREATE table training.fin(Id int Primary Key,Salary Int);

/*Insert values into Fin table*/
INSERT into training.fin(Id,Salary) values(2,200),(3,300),(4,400);

/*Check data in Emp table */
SELECT * from training.emp;

/*Check data in Fin table */
SELECT * from training.emp;
```

Output

Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim

Fin

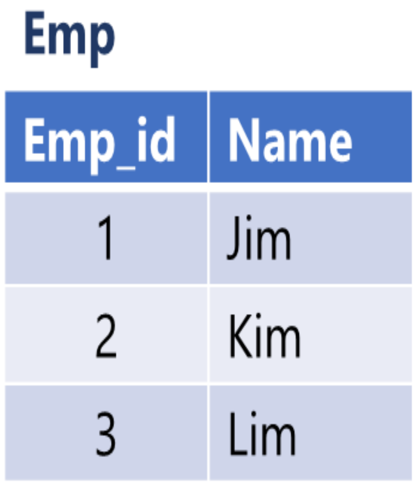
Id	Salary
2	200
3	300
4	400

Joins: LEFT Join (or LEFT Outer Join)

The LEFT JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

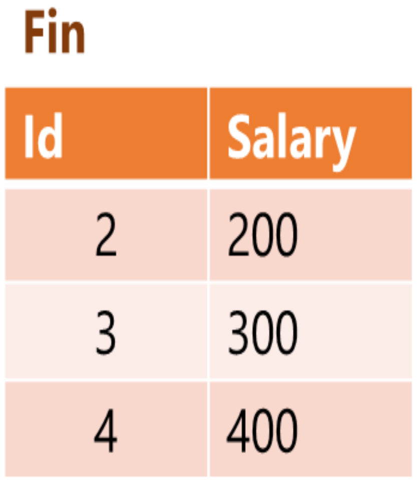
*/*LEFT Join query*/*

```
SELECT a.Emp_id,a.Name, b.Salary
FROM training.Emp a LEFT JOIN training.Fin b
ON a.Emp_id=b.Id;
```



Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim



Fin

Id	Salary
2	200
3	300
4	400

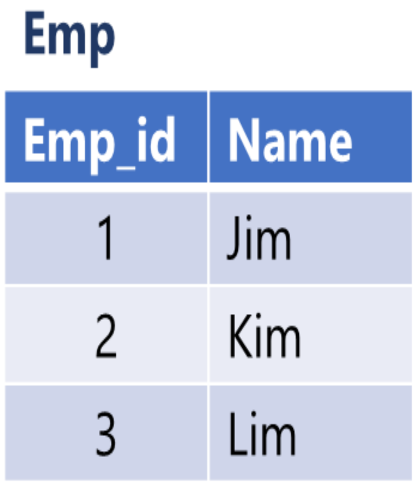
Left Join Output

Emp_id	Name	Salary
1	Jim	NULL
2	Kim	200
3	Lim	300

Joins: RIGHT Join (or Right Outer Join)

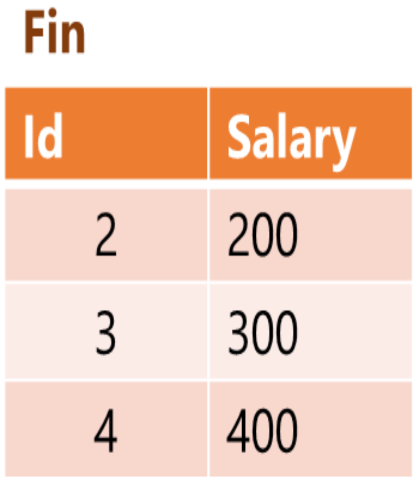
The RIGHT JOIN returns all rows from the right hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

```
/*Right Join query*/
SELECT b.id,a.Name, b.Salary
FROM training.Emp a RIGHT JOIN training.Fin b
ON a.Emp_id=b.Id;
```



Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim



Fin

Id	Salary
2	200
3	300
4	400

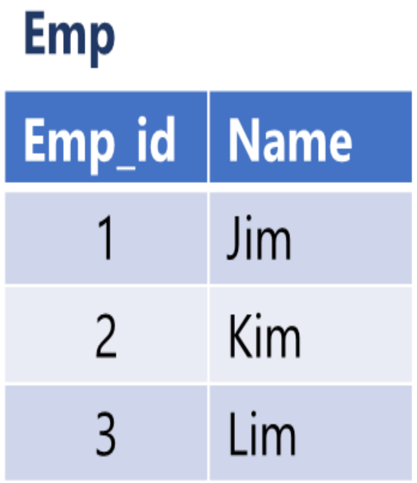
Right Join Output

Emp_id	Name	Salary
2	Kim	200
3	Lim	300
4	NULL	400

Joins: INNER Join (or Equi Join)

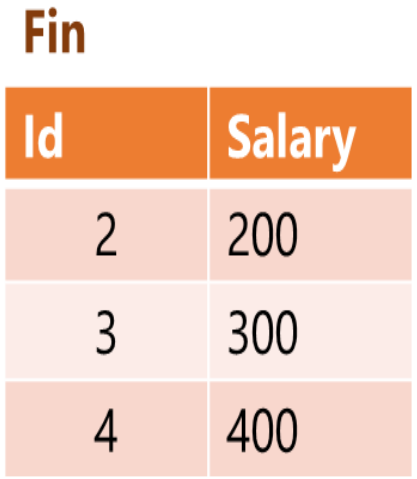
The INNER JOIN returns all rows only for the keys (IDs) that are available in both the tables.

```
/*Inner Join query*/
SELECT a.EMP_id,a.Name, b.Salary
FROM training.Emp a INNER JOIN training.Fin b
ON a.Emp_id=b.Id;
```



Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim



Fin

Id	Salary
2	200
3	300
4	400

Inner Join Output

Emp_id	Name	Salary
2	Kim	200
3	Lim	300

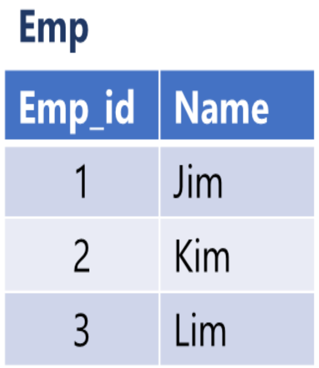
Joins: OUTER Join (or Full Outer Join)

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

```
/*Outer Join query for MySQL*/
SELECT a.Emp_id,a.Name, b.Salary
FROM training.Emp a LEFT JOIN training.Fin b
ON a.Emp_id=b.Id
UNION
SELECT b.id,a.Name, b.Salary
FROM training.Emp a RIGHT JOIN training.Fin b
ON a.Emp_id=b.Id;
```

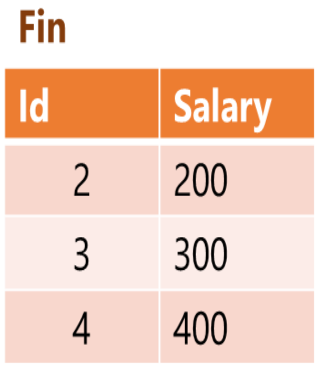
***NOTE:** MySQL doesn't support outer join and hence it can be achieved by using UNION of Left and Right Join.*

```
/* Outer join query for other platforms*/
SELECT a.Emp_id,a.Name, b.Salary
FROM training.Emp a OUTER JOIN training.Fin b
ON a.Emp_id=b.Id;
```



Emp

Emp_id	Name
1	Jim
2	Kim
3	Lim



Fin

Id	Salary
2	200
3	300
4	400

Outer Join Output

Emp_id	Name	Salary
1	Jim	NULL
2	Kim	200
3	Lim	300
4	NULL	400

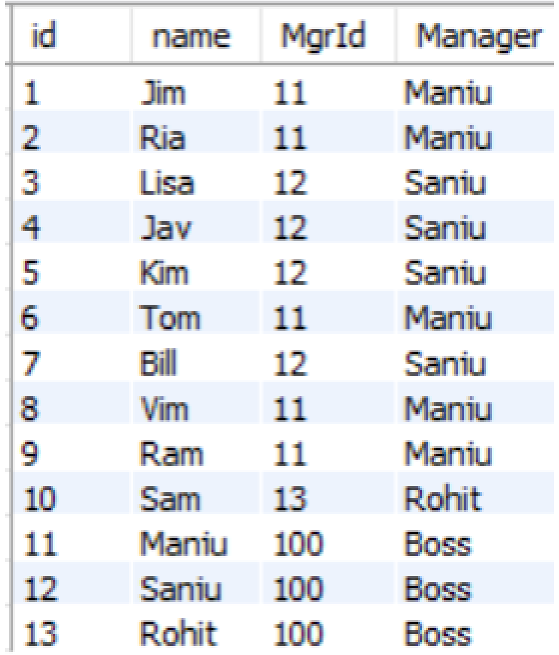
Joins: SELF Join

The SELF JOIN is basically joining the table with itself. To understand how it works and when to use it lets create this table.

```
/*Create the table Team*/
create table training.Team
(id int not null,
name varchar(20),
MgrId INT not null,
Manager varchar(20)
);
```

```
/*Insert data into the table Team*/
INSERT into training.Team
(id, name,MgrId, Manager)VALUES
(1,'Jim',11, 'Manju’),
(2,'Ria',11, 'Manju’),
(3,'Lisa',12, 'Sanju’),
(4,'Jay',12, 'Sanju’),
(5,'Kim',12, 'Sanju’),
(6,'Tom',11, 'Manju’),
(7,'Bill',12, 'Sanju’),
(8,'Vim',11, 'Manju’),
(9,'Ram',11, 'Manju’),
(10,'Sam',13, 'Rohit’),
(11,'Manju',100, 'Boss’),
(12,'Sanju',100, 'Boss’),
(13,'Rohit',100, 'Boss');
```

Training.Team



id	name	MgrId	Manager
1	Jim	11	Maniu
2	Ria	11	Maniu
3	Lisa	12	Saniu
4	Jav	12	Saniu
5	Kim	12	Saniu
6	Tom	11	Maniu
7	Bill	12	Saniu
8	Vim	11	Maniu
9	Ram	11	Maniu
10	Sam	13	Rohit
11	Maniu	100	Boss
12	Saniu	100	Boss
13	Rohit	100	Boss

Joins: SELF Join

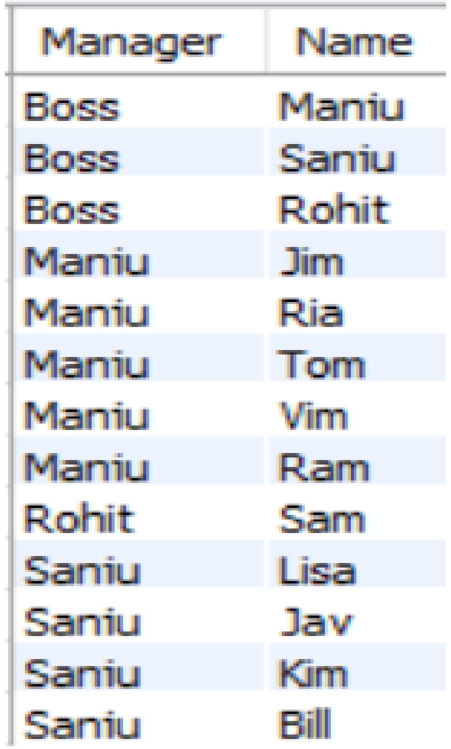
Now, let's assume that wee need to find team by manager (i.e. Manager name and direct reportee). Since both the data are in the same table, we need to perform self join to get this output.

```
/*SELF Join Query 1*/
Select a.Manager as Manager, b.Name AS Name
from training.team a INNER JOIN training.team b
on a.id=b.id
order by Manager;

/*SELF Join Query 2*/
Select a.Manager as Manager, b.Name AS Name
from training.team a, training.team b
Where a.id=b.id order by Manager;
```

Note: *Both the queries above will return same output.*

Self Join Output



Manager	Name
Boss	Maniu
Boss	Saniu
Boss	Rohit
Maniu	Jim
Maniu	Ria
Maniu	Tom
Maniu	Vim
Maniu	Ram
Rohit	Sam
Saniu	Lisa
Saniu	Jav
Saniu	Kim
Saniu	Bill

CASE Statement

CASE Statement

SQL CASE statement is like an “If-then-else condition”. It goes through conditions and returns a value when the condition is met. If it doesn’t find any true condition then it returns the value that is mentioned in the ELSE part. If there is no ELSE part and no conditions are true, it returns NULL.

Let’s understand it using the country data from world databases.

Scenario 1:

We know that we have country name and continent in the country table. Now we want to see the data by geography i.e. AMS, APAC, EMEA and OTHER. Note that the geography name is not available in the country data but we want to see the data by geography. Hence we are going to use CASE statement in our query to get the desired result.

```
SELECT country.name,country.Continent,
(CASE
WHEN Continent = 'North America' THEN 'AMS’
WHEN Continent = 'South America' THEN 'AMS’
WHEN Continent = 'Europe' THEN 'EMEA’
WHEN Continent = 'Africa' THEN 'EMEA'
WHEN Continent = 'Asia' THEN 'APAC'
ELSE 'OTHER’
END) AS Geo
```

Execute this query and check the output. You will see a new column by name “Geo” with AMS, APAC, EMEA and OTHER

We would love to hear back!

☎ **80737 99421**
86185 69998

✉ **info@introtallent.com**
www.introtallent.com

Office Address:
Introtallent Pvt Ltd.
#12, Anu Arcade, 3rd Floor
CMH Road, Indiranagar,
Bangalore – 560038