Three table JOIN syntax in SQL

Here is a general SQL query syntax to join three or more table. This SQL query should work in all major relation database e.g. MySQL, Oracle, Microsoft SQLServer, Sybase, and PostgreSQL:

```
SELECT t1.col, t3.col FROM table1 join table2 ON table1.primarykey = table2.foreignkey join table3 ON table2.primarykey = table3.foreignkey;
```

We first join table 1 and table 2 which produce a temporary table with combined data from table1 and table2, which is then joined to table3.

SQL Query to JOIN three tables in MySQL

In order to better understand the *joining of 3 tables in the SQL query* let's see an example. Consider the popular example of Employee and Department schema. In our case, we have used a link table called Register which links or relates both Employee to Department. The primary key of the Employee table (emp_id) is a foreign key in Register and similarly, the primary key of the Department table (dept_id) is a foreign key in Register table.

In order to *write an SQL query to print employee name and department name* alongside we need to **join 3 tables**. First JOIN statement will join Employee and Register and create a temporary table which will have dept_id as another column. Now second JOIN statement will join this temp table with Department table on dept_id to get the desired result.

```
mysql> SELECT * FROM Employee;
+--------+----------+--------+
| emp_id | emp_name | salary |
+--------+----------+--------+
| 1      | James    |   2000 |
| 2      | Jack     |   4000 |
| 3      | Henry    |   6000 |
| 4      | Tom      |   8000 |
+--------+----------+--------+
4 rows IN SET (0.00 sec)

mysql> SELECT * FROM Department;
+---------+-----------+
| dept_id | dept_name |
+---------+-----------+
| 101     | Sales     |
| 102     | Marketing |
| 103     | Finance   |
+---------+-----------+
3 rows IN SET (0.00 sec)

mysql> SELECT * FROM Register;
+--------+---------+
| emp_id | dept_id |
+--------+---------+
|      1 |     101 |
|      2 |     102 |
```

```
|      3 |     103 |
|      4 |     102 |
+--------+---------+
4 rows IN SET (0.00 sec)

mysql> SELECT emp_name, dept_name FROM Employee e JOIN Register r ON
e.emp_id=r.emp_id JOIN Department d ON r.dept_id=d.dept_id;
+----------+-----------+
| emp_name | dept_name |
+----------+-----------+
| James    | Sales     |
| Jack     | Marketing |
| Henry    | Finance   |
| Tom      | Marketing |
+----------+-----------+
4 rows IN SET (0.01 sec)
```

Joining three or more tables in SQL

There may occur some situations sometimes where data needs to be fetched from three or more tables. This article deals with two approaches to achieve it.

Example:
Creating three tables:

1. student
2. marks
3. details

Table 1: student

```
create table student(s_id int primary key,
                     s_name varchar(20));

insert into student values(1, 'Jack');
insert into student values(2, 'Rithvik');
insert into student values(3, 'Jaspreet');
insert into student values(4, 'Praveen');
insert into student values(5, 'Bisa');
insert into student values(6, 'Suraj');
```

Student

	s_id	s_name
1	1	Jack
2	2	Rithvik
3	3	Jaspreet
4	4	Praveen
5	5	Bisa
6	6	Suraj

Here s_id is the primary key

Table 2: marks

```
create table marks(school_id int primary key, s_id int,
                   score int, status varchar(20));

insert into marks values(1004, 1, 23, 'fail');
insert into marks values(1008, 6, 95, 'pass');
insert into marks values(1012, 2, 97, 'pass');
insert into marks values(1016, 7, 67, 'pass');
insert into marks values(1020, 3, 100, 'pass');
insert into marks values(1025, 8, 73, 'pass');
insert into marks values(1030, 4, 88, 'pass');
insert into marks values(1035, 9,  13, 'fail');
insert into marks values(1040, 5,  16, 'fail');
insert into marks values(1050, 10, 53, 'pass');
```

MARKS

	school_id	s_id	score	status
1	1004	1	23	fail
2	1008	6	95	pass
3	1012	2	97	pass
4	1016	7	67	pass
5	1020	3	100	pass
6	1025	8	73	pass
7	1030	4	88	pass
8	1035	9	13	fail
9	1040	5	16	fail
10	1050	10	53	pass

school_id is primary key marks table

Here s_id acts as a foreign key

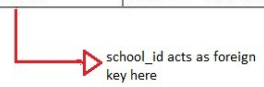
Table 3: details

```
create table details(address_city varchar(20), email_ID varchar(20),
                     school_id int, accomplishments varchar(50));
```

```
insert into details values('Banglore',  'jsingh@geeks.com',
                           1020, 'ACM ICPC selected');
insert into details values('Hyderabad', 'praveen@geeks.com',
                           1030, 'Geek of the month');
insert into details values('Delhi',     'rithvik@geeks.com',
                            1012, 'IOI finalist');
insert into details values('Chennai',   'om@geeks.com',
                           1111, 'Geek of the year');
insert into details values('Banglore', ' suraj@geeks.com',
                           1008, 'IMO finalist');
insert into details values('Mumbai',    'sasukeh@geeks.com',
                            2211, 'Made a robot');
insert into details values('Ahmedabad', 'itachi@geeks.com',
                           1172, 'Code Jam finalist');
insert into details values('Jaipur',    'kumar@geeks.com',
                            1972, 'KVPY finalist');
```

Details

	address_city	email_ID	school_id	accomplishments
1	Banglore	jsingh@geeks.com	1020	ACM ICPC selected
2	Hyderabad	praveen@geeks.com	1030	Geek of the month
3	Delhi	rithvik@geeks.com	1012	IOI finalist
4	Chennai	om@geeks.com	1111	Geek of the year
5	Banglore	suraj@geeks.com	1008	IMO finalist
6	Mumbai	sasukeh@geeks.com	2211	Made a robot
7	Ahmedabad	itachi@geeks.com	1172	Code Jam finalist
8	Jaipur	kumar@geeks.com	1972	KVPY finalist



Two approaches to join three or more tables:

1. Using joins in sql to join the table:

The same logic is applied which is done to join 2 tables i.e. **minimum** number of join statements to join **n** tables are **(n-1)**.

Query:

```
select s_name, score, status, address_city, email_id,
accomplishments from student s inner join marks m on
s.s_id = m.s_id inner join details d on
d.school_id = m.school_id;
```

Output:

	s_name	score	status	address_city	email_id	accomplishments
1	Jaspreet	100	pass	Banglore	jsingh@geeks.com	ACM ICPC selected
2	Praveen	88	pass	Hyderabad	praveen@geeks.com	Geek of the month
3	Rithvik	97	pass	Delhi	rithvik@geeks.com	IOI finalist
4	Suraj	95	pass	Banglore	suraj@geeks.com	IMO finalist

2. Using parent-child relationship:

This is rather an interesting approach. Create column **X** as primary key in one table and as foreign key in another table (i.e creating a parent-child relationship).

Let's look in the tables created:

s_id is the **primary key** in student table and is **foreign key** in marks table. **(student (parent) – marks(child))**.

school_id is the **primary key** in marks table and **foreign key** in details table. **(marks(parent) – details(child))**.

Query:

```
select s_name, score, status, address_city,
email_id, accomplishments from student s,
marks m, details d where s.s_id = m.s_id and
m.school_id = d.school_id;
```

Output:

	s_name	score	status	address_city	email_id	accomplishments
1	Jaspreet	100	pass	Banglore	jsingh@geeks.com	ACM ICPC selected
2	Praveen	88	pass	Hyderabad	praveen@geeks.com	Geek of the month
3	Rithvik	97	pass	Delhi	rithvik@geeks.com	IOI finalist
4	Suraj	95	pass	Banglore	suraj@geeks.com	IMO finalist