SQL joins is used to combine 2 or more tables based on the common column present in the tables.

There are different kinds of joins:

- > Inner Join
- Left Join
- > Right Join
- ➤ Full Join

We are going to perform the join operations on the mysql database. Hence, before we begin the discussion about the joins, You should have mysql database installed in your linux machine. If you don't have any idea about the installation and configuration of mysql DB, you can refer to the below mentioned article.

https://www.thegeekstuff.com/2008/07/howto-install-mysql-on-linux/

We will perform some join operations on retail\_db database which you will need to import to the mysql database. You can download the database dump from the below mentioned link:

https://raw.githubusercontent.com/dgadiraju/code/master/hadoop/edw/database/retail\_db.sql

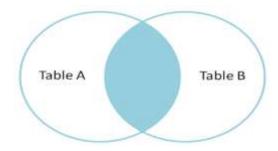
click on this link and do ctrl + S and save it as retail\_db.sql

First extract the sql file from the above mentioned package. Then you will have to import the sql dump. If you don't have any idea about how to import the data from sql dump into mysql, follow the below mentioned steps:

- First login to the mysql DB and create retial\_db database using command:
  CREATE DATABASE retail db;
- restore the data from the sql dump
  mysql -u root -p<root password> retail\_db < retail\_db.sql</pre>

Now, Let's start with Joins:

**Inner Join**: The inner join will join 2 or more tables as long as the condition satisfies. INNER JOIN or JOIN will provide the result set that matches both Table A and Table B.



#### Syntax:

```
SELECT <t1>.<column1>,<t1>.<column2>,<t1>.<column3>,<t2>.<column1>,<t2>.<column2>
FROM Table1 as t1, Table2 as t2
WHERE <t1>.<column2>;
```

OR

```
SELECT <t1>.<column1>,<t1>.<column2>,<t1>.<column3>,<t2>.<column1>,<t2>.<column2>
FROM Table1 as t1 INNER JOIN Table2 as t2
ON <t1>.<column2> = <t2>.<column2>;
```

Note: Instead of INNER JOIN, you can simply write JOIN and it will still work ©

#### Example:

```
mysql> show databases;
 Database
| information schema |
retail db
| training db
4 rows in set (0.00 sec)
mysql> use retail db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
| Tables_in_retail_db |
| categories
customers
departments
dummy
order_items
order items nopk
orders
products
8 rows in set (0.00 sec)
mysql>
mysql> describe categories;
                     | Type | Null | Key | Default | Extra
| Field
| category name | varchar(45) | NO |
                                              NULL
3 rows in set (0.00 sec)
mysql> describe departments;
        | Type | Null | Key | Default | Extra
Field
| department id | int(11) | NO | PRI | NULL | auto increment |
| department_name | varchar(45) | NO | | NULL
2 rows in set (0.00 sec)
```

```
mysql> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \
    -> FROM categories as c, departments as d \
    -> WHERE c.category_department_id = d.department_id;
```

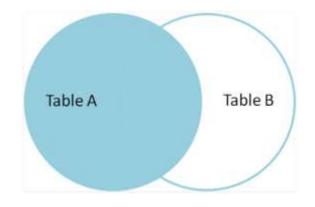
OR the same query can also be written using join statement as:

```
mysql> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \
    -> FROM categories as c JOIN departments as d \
```

<sup>-&</sup>gt; ON c.category\_department\_id = d.department\_id;

-> FROM cat	tegories as c,departments	as d \	_	d.department_name, c.category_r
-> WHERE C.	.category_department_id = +	+	, +	++
ategory_id	category_department_id	department_id	department_nam	e   category_name
1	2	   2	Fitness	Football
2	2	2	Fitness	Soccer
3	] 2	2	Fitness	Baseball & Softball
4	2	2	Fitness	Basketball
5	] 2	2	Fitness	Lacrosse
6	2	2	Fitness	Tennis & Racquet
7	2	2	Fitness	Hockey
8	2	2	Fitness	More Sports
9	] 3	] 3	Footwear	Cardio Equipment
10	] 3	] 3	Footwear	Strength Training
11	] 3	3	Footwear	Fitness Accessories
12	] 3	3	Footwear	Boxing & MMA
13	] 3	] 3	Footwear	Electronics
14	] 3	] 3	Footwear	Yoga & Pilates
15	] 3	] 3	Footwear	Training by Sport
16	] 3	] 3	Footwear	As Seen on TV!
17	l 4		Apparel	Cleats
18	4	4	Apparel	Men's Footwear
19	l 4		Apparel	Women's Footwear
20	4	4	Apparel	Kids' Footwear
21	l 4	4	Apparel	Featured Shops
22	l 4	4	Apparel	Accessories
23	[ 5	5	Golf	Men's Apparel
24	[ 5	5	Golf	Women's Apparel
25	[ 5	5	Golf	Boys' Apparel
26	5		Golf	Girls' Apparel
27		•	Golf	Accessories
28			Golf	Top Brands
29	•		Golf	Shop By Sport
30 31	6	j 6	Outdoors   Outdoors	Men's Golf Clubs   Women's Golf Clubs
32			Outdoors	Golf Apparel
33		•	Outdoors	Golf Shoes
34			Outdoors	Golf Bags & Carts
35			Outdoors	Golf Gloves
36			Outdoors	Golf Balls
37			Outdoors	Electronics
38			Outdoors	Kids' Golf Clubs
39			Outdoors	Team Shop
40			Outdoors	Accessories
41			Outdoors   Fan Shop	Trade-In
42		•		Bike & Skate Shop
43			Fan Shop   Fan Shop	Camping & Hiking
44			Fan Shop   Fan Shop	Hunting & Shooting
45 46			Fan Shop   Fan Shop	Fishing     Indoor/Outdoor Games
47			Fan Shop   Fan Shop	
48		•	Fan Shop   Fan Shop	Boating     Water Sports

**Left Join :** This join will return all the rows from the table on the left side of the join and matching rows from the table on the right side of the join. The rows for which there are no matching rows on the right side, the result will contain NULL. LEFT JOIN is also called as LEFT OUTER JOIN



# Syntax:

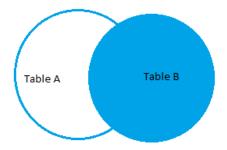
SELECT <t1>.<column1>,<t1>.<column2>,<t1>.<column3>,<t2>.<column1>,<t2>.<column2> FROM Table1 as t1 LEFT JOIN Table2 as t2 ON  $\langle t1 \rangle$ . $\langle column2 \rangle = \langle t2 \rangle$ . $\langle column2 \rangle$ ;

## Example:

mysql> SELECT c.category\_id, c.category\_department\_id, d.department\_id,d.department\_name, c.category\_name \
 -> FROM categories as c LEFT JOIN departments as d \
 -> ON c.category\_department\_id = d.department\_id;

<pre>mysql&gt; SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \</pre>						
-> ON c.category_department_id = d.department_id;						
+   category_id	+   category_department_id	+   department_id	department_name	++   category_name		
1	   2	2	Fitness	Football		
2	2	2	Fitness	Soccer		
3	] 2			Baseball & Softball		
1 4	2			Basketball		
] 5     6	] 2 ] 2			Lacrosse     Tennis & Racquet		
7	2			Hockey		
8	] 2	2	Fitness	More Sports		
9	] 3			Cardio Equipment		
10   11	] ] 3			Strength Training     Fitness Accessories		
1 12	] 3			Boxing & MMA		
13	] 3			Electronics		
14	] 3	3	Footwear	Yoga & Pilates		
15	] 3			Training by Sport		
16   17	] 3   4			As Seen on TV!     Cleats		
1 18	I 4			Cleats   Men's Footwear		
19	4			Women's Footwear		
20	] 4	4	Apparel	Kids' Footwear		
21	1 4			Featured Shops		
] 22 ] 23	4   5			Accessories   Men's Apparel		
23	] 5			Women's Apparel		
25				Boys' Apparel		
26	] 5	5	Golf	Girls' Apparel		
27	5			Accessories		
28	5			Top Brands		
] 29 ] 30	] 5   6			Shop By Sport     Men's Golf Clubs		
31	6			Women's Golf Clubs		
32	6	[ 6	Outdoors	Golf Apparel		
33	. 6			Golf Shoes		
] 34 ] 35	6   6			Golf Bags & Carts     Golf Gloves		
1 36	I 6			Golf Balls		
37	6			Electronics		
38	] 6	[ 6	Outdoors	Kids' Golf Clubs		
39	J 6	[ 6 ]	Outdoors	Team Shop		
40	J 6			Accessories		
41	[ 6 [ 7			Trade-In     Bike & Skate Shop		
42     43			Fan Shop Fan Shop	Bike & Skate Shop     Camping & Hiking		
44				Hunting & Shooting		
45			Fan Shop	Fishing		
46			Fan Shop	Indoor/Outdoor Games		
47   48			Fan Shop Fan Shop	Boating     Water Sports		
49				MLB		
50				NFL		
51				NHL		
52				NBA		
53   54				NCAA     MLS		
55				MLS     International Soccer		
56				World Cup Shop		
57				MLB Players		
58	J 8	NULL	NULL	NFL Players		
58 rows in set (0.00 sec)						
mysql>						

**Right Join:** This join will return all the rows from the table on the right side of the join and matching rows from the table on the left side of the join. The rows for which there are no matching rows on the left side, the result will contain NULL. RIGHT JOIN is also called as RIGHT OUTER JOIN



## Syntax:

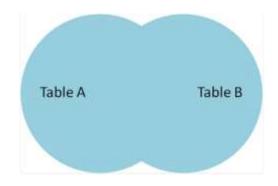
```
SELECT <t1>.<column1>,<t1>.<column2>,<t1>.<column3>,<t2>.<column1>,<t2>.<column2>
FROM Table1 as t1 RIGHT JOIN Table2 as t2
ON <t1>.<column2> = <t2>.<column2>;
```

## Example:

```
mysql> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \
    -> FROM categories as c RIGHT JOIN departments as d \
    -> ON c.category_department_id = d.department_id;
```

-> ON c.category_department_id = d.department_id;						
gory_id	,   category_department_id +	department_id   +	department_name	category_name		
1	2	2	Fitness	Football		
2	] 2	2	Fitness	Soccer		
3	] 2	2	Fitness	Baseball & Softball		
4	] 2			Basketball		
5	2	2	Fitness	Lacrosse		
6	2		Fitness	Tennis & Racquet		
7	2			Hockey		
8	2	2	Fitness	More Sports		
9	] 3			Cardio Equipment		
10	] 3	] 3	Footwear	Strength Training		
11	] 3	] 3	Footwear	Fitness Accessories		
12	] 3	3	Footwear	Boxing & MMA		
13	] 3	3	Footwear	Electronics		
14	] 3	] 3	Footwear	Yoga & Pilates		
15	] 3	] 3	Footwear	Training by Sport		
16	] 3	] 3	Footwear	As Seen on TV!		
17	4	4	Apparel	Cleats		
18	4		Apparel	Men's Footwear		
19	4			Women's Footwear		
20	4	4	Apparel	Kids' Footwear		
21	4			Featured Shops		
22	4	4	Apparel	Accessories		
23	5	5	Golf	Men's Apparel		
24	5	5	Golf	Women's Apparel		
25	5	5	Golf	Boys' Apparel		
26	5	5	Golf	Girls' Apparel		
27	5	5	Golf	Accessories		
28	5	5	Golf	Top Brands		
29	5	5	Golf	Shop By Sport		
30	[ 6	[ 6 ]	Outdoors	Men's Golf Clubs		
31	[ 6			Women's Golf Clubs		
32	6	6	Outdoors	Golf Apparel		
33	[ 6	[ 6	Outdoors	Golf Shoes		
34	6	[ 6 ]	Outdoors	Golf Bags & Carts		
35	[ 6	[ 6	Outdoors	Golf Gloves		
36	6	6	Outdoors	Golf Balls		
37	6	[ 6 ]	Outdoors	Electronics		
38	[ 6	1 6	Outdoors	Kids' Golf Clubs		
39	1	6	Outdoors	Team Shop		
40	[ 6	1 6	Outdoors	Accessories		
41			Outdoors	Trade-In		
42	1 7	7	Fan Shop	Bike & Skate Shop		
43	1 7	7	Fan Shop	Camping & Hiking		
44	1 7	7	Fan Shop	Hunting & Shooting		
45	7	7	Fan Shop	Fishing		
46	1 7	7	Fan Shop	Indoor/Outdoor Games		
47	1 7	7	Fan Shop	Boating		
48	7	7	Fan Shop	Water Sports		

**Full Join:** This join will return the result set by combining both the LEFT JOIN and RIGHT JOIN. The result set will contain all the rows from both the tables. The rows for which there are no matching, the result set will contain NULL values



## Syntax:

```
SELECT <t1>.<column1>,<t1>.<column2>,<t1>.<column3>,<t2>.<column1>,<t2>.<column2>
FROM Table1 as t1 FULL JOIN Table2 as t2
ON <t1>.<column2> = <t2>.<column2>;
```

The above syntax can be used for other SQL databases. MySQL does not support FULL JOIN. Hence can implement the same by using UNION on LEFT JOIN and RIGHT JOIN.

## Example:

```
mysql> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \
    -> FROM categories as c LEFT JOIN departments as d ON c.category_department_id = d.department_id \
    -> UNION
    -> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \
```

-> FROM categories as c RIGHT JOIN departments as d ON c.category\_department\_id = d.department\_id ;

mysql> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \ -> FROM categories as c LEFT JOIN departments as d ON c.category_department id = d.department id \						
-> UNION						
-> SELECT c.category_id, c.category_department_id, d.department_id,d.department_name, c.category_name \ -> FROM categories as c RIGHT JOIN departments as d ON c.category_department_id = d.department_id;						
+	+	+	+	+	+	
+	category_department_id   +	department_1d	+	category_name +	·+	
1			Fitness	Football	!	
] 2	] ] 2		Fitness   Fitness	Soccer   Baseball & Softball		
4	2			Basketball	i	
5	2	2	Fitness	Lacrosse	İ	
6	2			Tennis & Racquet	1	
] 7	] ] 2			Hockey	!	
1 8	] 2			More Sports   Cardio Equipment		
10				Strength Training	i	
11	3	3		Fitness Accessories	i	
12	] 3	3	Footwear	Boxing & MMA	I	
13				Electronics	1	
14				Yoga & Pilates		
15   16	] ] 3			Training by Sport   As Seen on TV!		
17	1 4			Cleats		
18	1 4			Men's Footwear	i	
19	4			Women's Footwear	i	
20	4	4	Apparel	Kids' Footwear	I .	
21				Featured Shops	1	
22				Accessories	!	
] 23 ] 24				Men's Apparel   Women's Apparel	!	
25	] 5			Boys' Apparel		
1 26				Girls' Apparel	i	
27	5	5		Accessories	i	
28	[ 5	5	Golf	Top Brands	I	
29	5			Shop By Sport	1	
30				Men's Golf Clubs	!	
] 31 ] 32	[ 6   [ 6			Women's Golf Clubs   Golf Apparel		
1 33	6			Golf Shoes		
34	6			Golf Bags & Carts	i	
35	[ 6	6	Outdoors	Golf Gloves	T.	
36	[ 6	6	Outdoors	Golf Balls	I	
37				Electronics		
38			Outdoors	Kids' Golf Clubs		
] 39     40				Team Shop   Accessories		
41				Trade-In		
42				Bike & Skate Shop		
43		7		Camping & Hiking		
44				Hunting & Shooting		
45				Fishing		
46     47				Indoor/Outdoor Games   Boating		
48				Water Sports		
49		NULL		MLB		
50		NULL		NFL		
51		NULL		NHL		
52		NULL		NBA		
53     54		NULL   NULL		NCAA   MLS		
55		NULL		International Soccer		
56		NULL		World Cup Shop		
57		NULL		MLB Players		
58	8	NULL	NULL	NFL Players		
58 rows in set (0.01 sec)						

Author - Shivshankar Chandankhede References : Thegeekstuff.com , geeksforgeeks.org, https://dev.mysql.com/doc/refman/8.0/en/join.html