**sudo mysql -u root -pRoot@123**

**CREATE DATABASE testDB;**

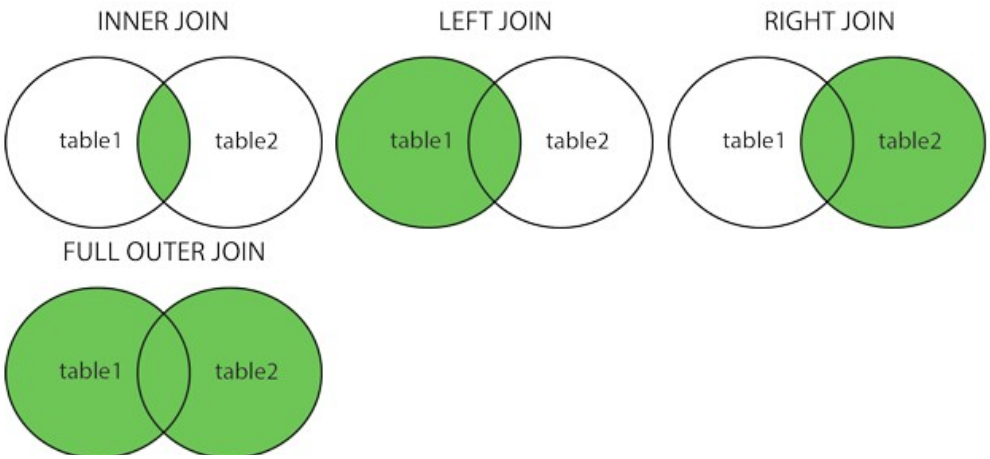
**use testDB;**

**SQL Join Operators:**
**o Cross Join**
**o Natural Join**
**o Join USING Clause**
**o Join ON Clause**
**o Outer Join**

# Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN` : Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN` : Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN` : Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN` : Returns all records when there is a match in either left or right table

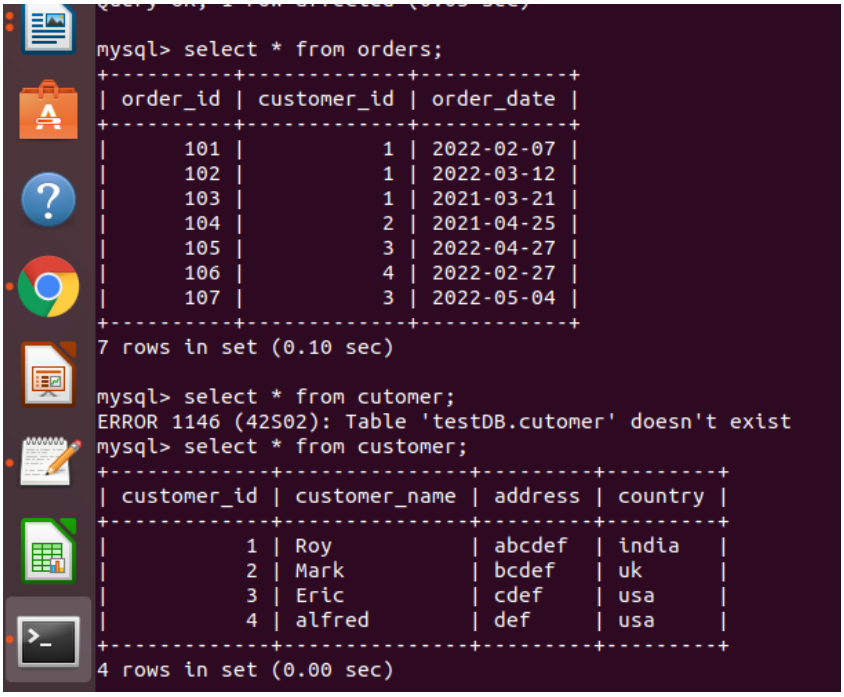


**create table customer (customer_id int, customer_name varchar(15), address varchar(15), country varchar(15), primary key(customer_id));**

**insert into customer values(1, 'Roy', 'abcdef', 'india');**

**create table orders (order_id int, customer_id int, order_date date, foreign key(customer_id) references customer(customer_id));**

**insert into orders values(101, 1, '2022-2-7');**



```
mysql> select * from orders;
+----------+-------------+------------+
| order_id | customer_id | order_date |
+----------+-------------+------------+
|      101 |           1 | 2022-02-07 |
|      102 |           1 | 2022-03-12 |
|      103 |           1 | 2021-03-21 |
|      104 |           2 | 2021-04-25 |
|      105 |           3 | 2022-04-27 |
|      106 |           4 | 2022-02-27 |
|      107 |           3 | 2022-05-04 |
+----------+-------------+------------+
7 rows in set (0.10 sec)

mysql> select * from cutomer;
ERROR 1146 (42S02): Table 'testDB.cutomer' doesn't exist
mysql> select * from customer;
+-------------+---------------+---------+---------+
| customer_id | customer_name | address | country |
+-------------+---------------+---------+---------+
|           1 | Roy           | abcdef  | india   |
|           2 | Mark          | bcdef   | uk      |
|           3 | Eric          | cdef    | usa     |
|           4 | alfred        | def     | usa     |
+-------------+---------------+---------+---------+
4 rows in set (0.00 sec)
```

**Natural Join**

**When we combine rows of two or more tables based on a common column between them, this operation is called joining. A natural join is a type of join operation that creates an implicit join by combining tables based on columns with the same name and data type. It is similar to the INNER or LEFT JOIN, but we cannot use the ON or USING clause with natural join as we used in them.**

**Points to remember:**
**There is no need to specify the column names to join.**
**The resultant table always contains unique columns.**
**It is possible to perform a natural join on more than two tables.**
**Do not use the ON clause.**

**Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.**

**The MySQL NATURAL JOIN is structured in such a way that, columns with the same name of associate tables will appear once only.**

**Natural Join Syntax is,**

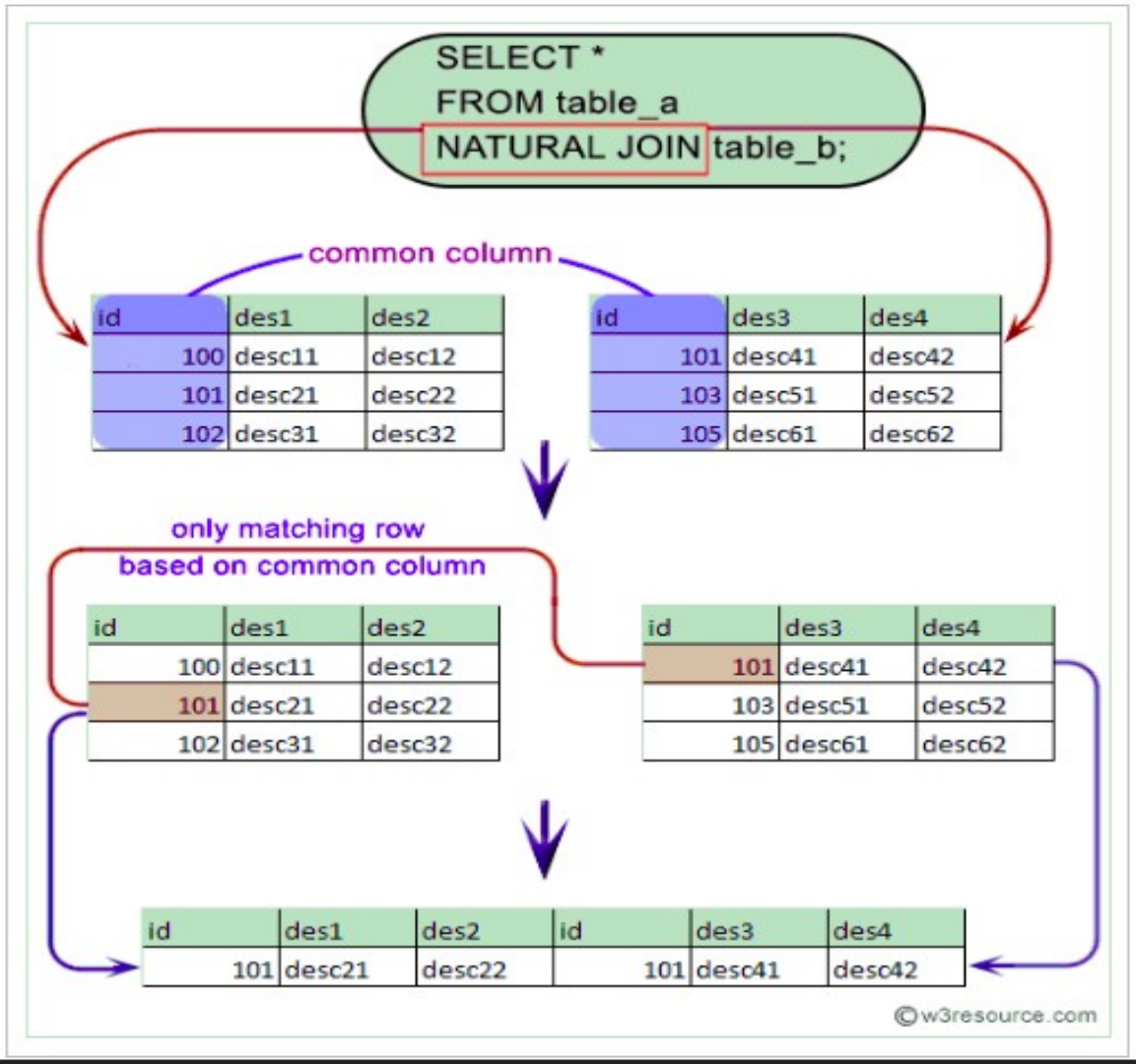
**SELECT * from table-name1 NATURAL JOIN table-name2;**

**SELECT * from customer NATURAL JOIN orders;**

```
+-------------+---------------+---------+---------+----------+------------+
28 rows in set (0.00 sec)

mysql> SELECT * from customer NATURAL JOIN orders;
+-------------+---------------+---------+---------+----------+------------+
| customer_id | customer_name | address | country | order_id | order_date |
+-------------+---------------+---------+---------+----------+------------+
|           1 | Roy           | abcdef  | india   |      101 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      104 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      105 | 2022-04-27 |
|           3 | Eric          | cdef    | usa     |      107 | 2022-05-04 |
|           4 | alfred        | def     | usa     |      106 | 2022-02-27 |
+-------------+---------------+---------+---------+----------+------------+
7 rows in set (0.30 sec)

mysql>
```

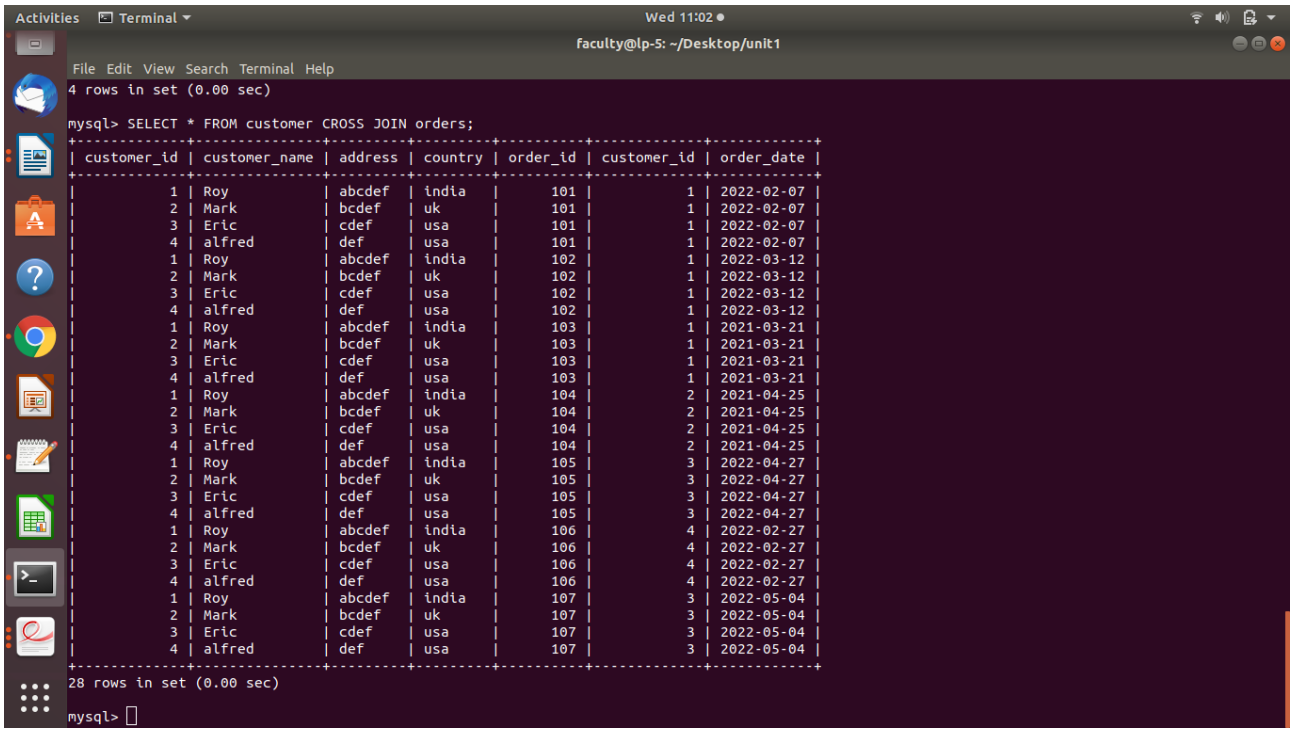

SELECT *
FROM table_a
NATURAL JOIN table_b;
common column
id des1 des2
100 desc11 desc12
101 desc21 desc22
102 desc31 desc32
id des3 des4
101 desc41 desc42
103 desc51 desc52
105 desc61 desc62
only matching row
based on common column
id des1 des2
100 desc11 desc12
101 desc21 desc22
102 desc31 desc32
id des3 des4
101 desc41 desc42
103 desc51 desc52
105 desc61 desc62
id des1 des2 id des3 des4
101 desc21 desc22 101 desc41 desc42
©w3resource.com

# Cross Join

**Cross JOIN is a simplest form of JOINs which matches each row from one database table to all rows of another.**

**In other words it gives us combinations of each row of first table with all records in second table.**

**mysql> SELECT * FROM customer CROSS JOIN orders;**



```
4 rows in set (0.00 sec)

mysql> SELECT * FROM customer CROSS JOIN orders;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           2 | Mark          | bcdef   | uk      |      101 |           1 | 2022-02-07 |
|           3 | Eric          | cdef    | usa     |      101 |           1 | 2022-02-07 |
|           4 | alfred        | def     | usa     |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           2 | Mark          | bcdef   | uk      |      102 |           1 | 2022-03-12 |
|           3 | Eric          | cdef    | usa     |      102 |           1 | 2022-03-12 |
|           4 | alfred        | def     | usa     |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      103 |           1 | 2021-03-21 |
|           3 | Eric          | cdef    | usa     |      103 |           1 | 2021-03-21 |
|           4 | alfred        | def     | usa     |      103 |           1 | 2021-03-21 |
|           1 | Roy           | abcdef  | india   |      104 |           2 | 2021-04-25 |
|           2 | Mark          | bcdef   | uk      |      104 |           2 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      104 |           2 | 2021-04-25 |
|           4 | alfred        | def     | usa     |      104 |           2 | 2021-04-25 |
|           1 | Roy           | abcdef  | india   |      105 |           3 | 2022-04-27 |
|           2 | Mark          | bcdef   | uk      |      105 |           3 | 2022-04-27 |
|           3 | Eric          | cdef    | usa     |      105 |           3 | 2022-04-27 |
|           4 | alfred        | def     | usa     |      105 |           3 | 2022-04-27 |
|           1 | Roy           | abcdef  | india   |      106 |           4 | 2022-02-27 |
|           2 | Mark          | bcdef   | uk      |      106 |           4 | 2022-02-27 |
|           3 | Eric          | cdef    | usa     |      106 |           4 | 2022-02-27 |
|           4 | alfred        | def     | usa     |      106 |           4 | 2022-02-27 |
|           1 | Roy           | abcdef  | india   |      107 |           3 | 2022-05-04 |
|           2 | Mark          | bcdef   | uk      |      107 |           3 | 2022-05-04 |
|           3 | Eric          | cdef    | usa     |      107 |           3 | 2022-05-04 |
|           4 | alfred        | def     | usa     |      107 |           3 | 2022-05-04 |
+-------------+---------------+---------+---------+----------+-------------+------------+
28 rows in set (0.00 sec)

mysql>
```

**INNER JOIN**

**The MySQL INNER JOIN is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of join.**

**Syntax :**
**SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;**
**OR**
**SELECT columns FROM table1 , table2 WHERE table1.column = table2.column**

**select * from customer INNER JOIN orders on customer.customer_id=orders.customer_id;**

```
mysql> select * from customer INNER JOIN orders on customer.customer_id=orders.customer_id;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      104 |           2 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      105 |           3 | 2022-04-27 |
|           3 | Eric          | cdef    | usa     |      107 |           3 | 2022-05-04 |
|           4 | alfred        | def     | usa     |      106 |           4 | 2022-02-27 |
+-------------+---------------+---------+---------+----------+-------------+------------+
7 rows in set (0.08 sec)

mysql>
```

**select * from customer INNER JOIN orders where customer.customer_id=orders.customer_id;**

**SELECT * FROM customer , orders WHERE customer.customer_id = orders.customer_id;**

**create table officers (officer_id int, officer_name varchar(15), address varchar(15));**

**insert into officers values(1, 'Roy', 'abcdef');**

**create table student (student_id int, student_name varchar(15), course_name varchar(15));**

**insert into student values(1, 'Alina', 'BCA');**

**show tables;**

```
+-----------------+
| Tables_in_testDB |
+-----------------+
| officers        |
| student         |
```

**mysql> select student.student_id,student.student_name,officers.officer_name from student inner join officers on student.student_id=officers.officer_id;**

```
+------------+--------------+--------------+
| student_id | student_name | officer_name |
+------------+--------------+--------------+
|          1 | Alina        | Roy          |
|          2 | Alexa        | Ankit        |
|          3 | mark         | Salman       |
+------------+--------------+--------------+
3 rows in set (0.00 sec)
```

**SELECT officers.officer_name, officers.address, student.course_name FROM officers INNER JOIN student ON officers.officer_id = student.student_id;**

**Left Outer Join**

**The LEFT OUTER 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.**

**The left outer join returns a result table with the matched data of two tables then remaining rows of the left table and null for the right table's column.**

**SELECT * FROM customer LEFT JOIN orders ON customer.customer_id=orders.customer_id;**

```
mysql> SELECT * FROM customer LEFT JOIN orders ON customer.customer_id=orders.customer_id;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      104 |           2 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      105 |           3 | 2022-04-27 |
|           4 | alfred        | def     | usa     |      106 |           4 | 2022-02-27 |
|           3 | Eric          | cdef    | usa     |      107 |           3 | 2022-05-04 |
|           6 | Amit          | dsdfs   | india   |     NULL |        NULL | NULL       |
+-------------+---------------+---------+---------+----------+-------------+------------+
8 rows in set (0.00 sec)

mysql>
```

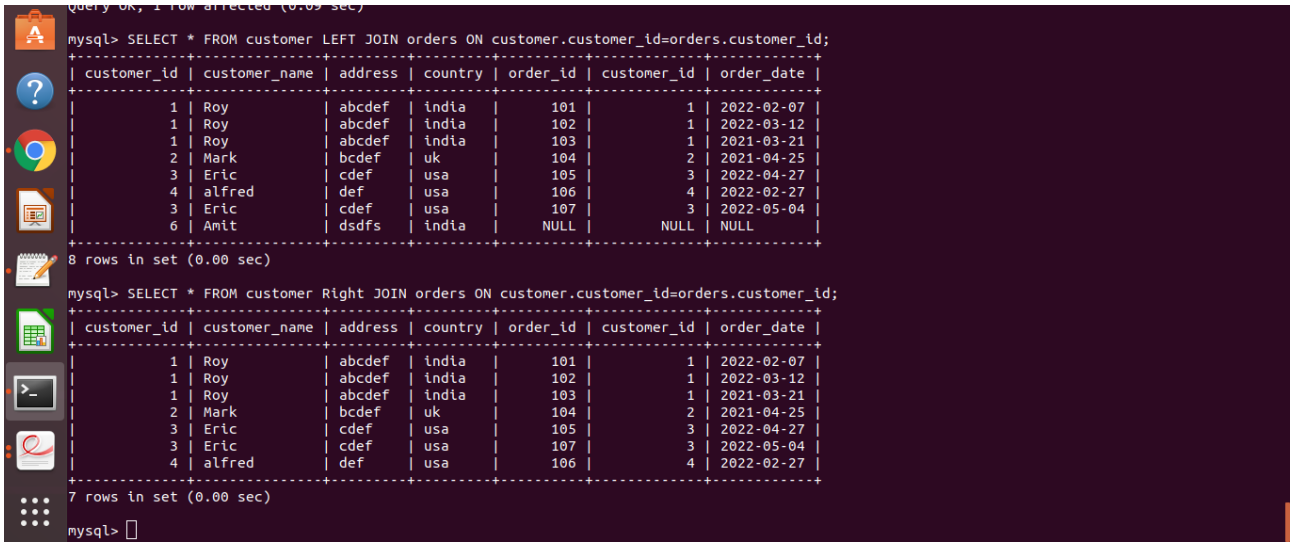
```
mysql> select * from orders;
+----------+-------------+------------+
| order_id | customer_id | order_date |
+----------+-------------+------------+
|      101 |           1 | 2022-02-07 |
|      102 |           1 | 2022-03-12 |
|      103 |           1 | 2021-03-21 |
|      104 |           2 | 2021-04-25 |
|      105 |           3 | 2022-04-27 |
|      106 |           4 | 2022-02-27 |
|      107 |           3 | 2022-05-04 |
+----------+-------------+------------+
7 rows in set (0.10 sec)

mysql> select * from cutomer;
ERROR 1146 (42S02): Table 'testDB.cutomer' doesn't exist
mysql> select * from customer;
+-------------+---------------+---------+---------+
| customer_id | customer_name | address | country |
+-------------+---------------+---------+---------+
|           1 | Roy           | abcdef  | india   |
|           2 | Mark          | bcdef   | uk      |
|           3 | Eric          | cdef    | usa     |
|           4 | alfred        | def     | usa     |
+-------------+---------------+---------+---------+
4 rows in set (0.00 sec)
```

## Right Join

**The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where he join condition is fulfilled.**

**SELECT * FROM customer RIGHT JOIN orders ON customer.customer_id=orders.customer_id;**



```
mysql> SELECT * FROM customer LEFT JOIN orders ON customer.customer_id=orders.customer_id;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      104 |           2 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      105 |           3 | 2022-04-27 |
|           4 | alfred        | def     | usa     |      106 |           4 | 2022-02-27 |
|           3 | Eric          | cdef    | usa     |      107 |           3 | 2022-05-04 |
|           6 | Amit          | dsdfs   | india   |     NULL |        NULL | NULL       |
+-------------+---------------+---------+---------+----------+-------------+------------+
8 rows in set (0.00 sec)

mysql> SELECT * FROM customer Right JOIN orders ON customer.customer_id=orders.customer_id;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
|           2 | Mark          | bcdef   | uk      |      104 |           2 | 2021-04-25 |
|           3 | Eric          | cdef    | usa     |      105 |           3 | 2022-04-27 |
|           3 | Eric          | cdef    | usa     |      107 |           3 | 2022-05-04 |
|           4 | alfred        | def     | usa     |      106 |           4 | 2022-02-27 |
+-------------+---------------+---------+---------+----------+-------------+------------+
7 rows in set (0.00 sec)

mysql>
```

**FULL JOIN**

**The full outer join returns a result table with the matched data of two table then remaining rows of both left table and then the right table.**

**SELECT * FROM customer**
**LEFT JOIN orders ON customer.customer_id = orders.customer_id**
**UNION**
**SELECT * FROM customer**
**RIGHT JOIN orders ON customer.customer_id = orders.customer_id**

**The ON clause**

**The ON clause is used to join tables where the column names don't match in both tables.**

**The join conditions are removed from the filter conditions in the WHERE clause.**

**select * from customer INNER JOIN orders on customer.customer_id=orders.customer_id where customer.customer_id=1;**

```
mysql> select * from customer INNER JOIN orders on customer.customer_id=orders.customer_id where customer.customer_id=1;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
+-------------+---------------+---------+---------+----------+-------------+------------+
3 rows in set (0.08 sec)
```

**The USING clause**

**The USING clause is used if several columns share the same name but you don't want to join using all of these common columns. The columns listed in the USING clause can't have any qualifiers in the statement, including the WHERE clause.**

**USING**

**The USING clause is something we don't need to mention in the JOIN condition when we are retrieving data from multiple tables. When we use USING clause, that particular column name should be present in both tables, and the SELECT query will automatically join those tables using the given column name in USING clause.**

**for e.g. if there are two common column names in the table, then Mention the desired common column name in the USING clause**

- **The USING clause: This allows you to specify the join key by name.**
- **The ON clause: This syntax allows you to specify the column names for join keys in both tables.**

**select * from customer INNER JOIN orders using(customer_id) where customer_id=1;**

```
3 rows in set (0.08 sec)

mysql> select * from customer INNER JOIN orders on customer.customer_id=orders.customer_id where customer.customer_id=1;
+-------------+---------------+---------+---------+----------+-------------+------------+
| customer_id | customer_name | address | country | order_id | customer_id | order_date |
+-------------+---------------+---------+---------+----------+-------------+------------+
|           1 | Roy           | abcdef  | india   |      101 |           1 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 |           1 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 |           1 | 2021-03-21 |
+-------------+---------------+---------+---------+----------+-------------+------------+
3 rows in set (0.00 sec)

mysql> select * from customer INNER JOIN orders using(customer_id) where customer_id=1;
+-------------+---------------+---------+---------+----------+------------+
| customer_id | customer_name | address | country | order_id | order_date |
+-------------+---------------+---------+---------+----------+------------+
|           1 | Roy           | abcdef  | india   |      101 | 2022-02-07 |
|           1 | Roy           | abcdef  | india   |      102 | 2022-03-12 |
|           1 | Roy           | abcdef  | india   |      103 | 2021-03-21 |
+-------------+---------------+---------+---------+----------+------------+
3 rows in set (0.00 sec)

mysql>
```