

SQL - EXCEPT CLAUSE

<http://www.tutorialspoint.com/sql/sql-except-clause.htm>

Copyright © tutorialspoint.com

The SQL **EXCEPT** clause/operator is used to combine two **SELECT** statements and returns rows from the first **SELECT** statement that are not returned by the second **SELECT** statement. This means **EXCEPT** returns only rows which are not available in second **SELECT** statement.

Just as with the **UNION** operator, the same rules apply when using the **EXCEPT** operator. MySQL does not support **EXCEPT** operator.

Syntax:

The basic syntax of **INTERSECT** is as follows:

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]

EXCEPT

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

Here given condition could be any given expression based on your requirement.

Example:

Consider following two tables, (a) **CUSTOMERS** table is as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

(b) Another table is **ORDERS** as follows:

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

Now let us join these two tables in our **SELECT** statement as follows:

```
SQL> SELECT ID, NAME, AMOUNT, DATE
      FROM CUSTOMERS
      LEFT JOIN ORDERS
      ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID
EXCEPT
```

```
SELECT ID, NAME, AMOUNT, DATE
FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

This would produce following result:

ID	NAME	AMOUNT	DATE
1	Ramesh	NULL	NULL
5	Hardik	NULL	NULL
6	Komal	NULL	NULL
7	Muffy	NULL	NULL