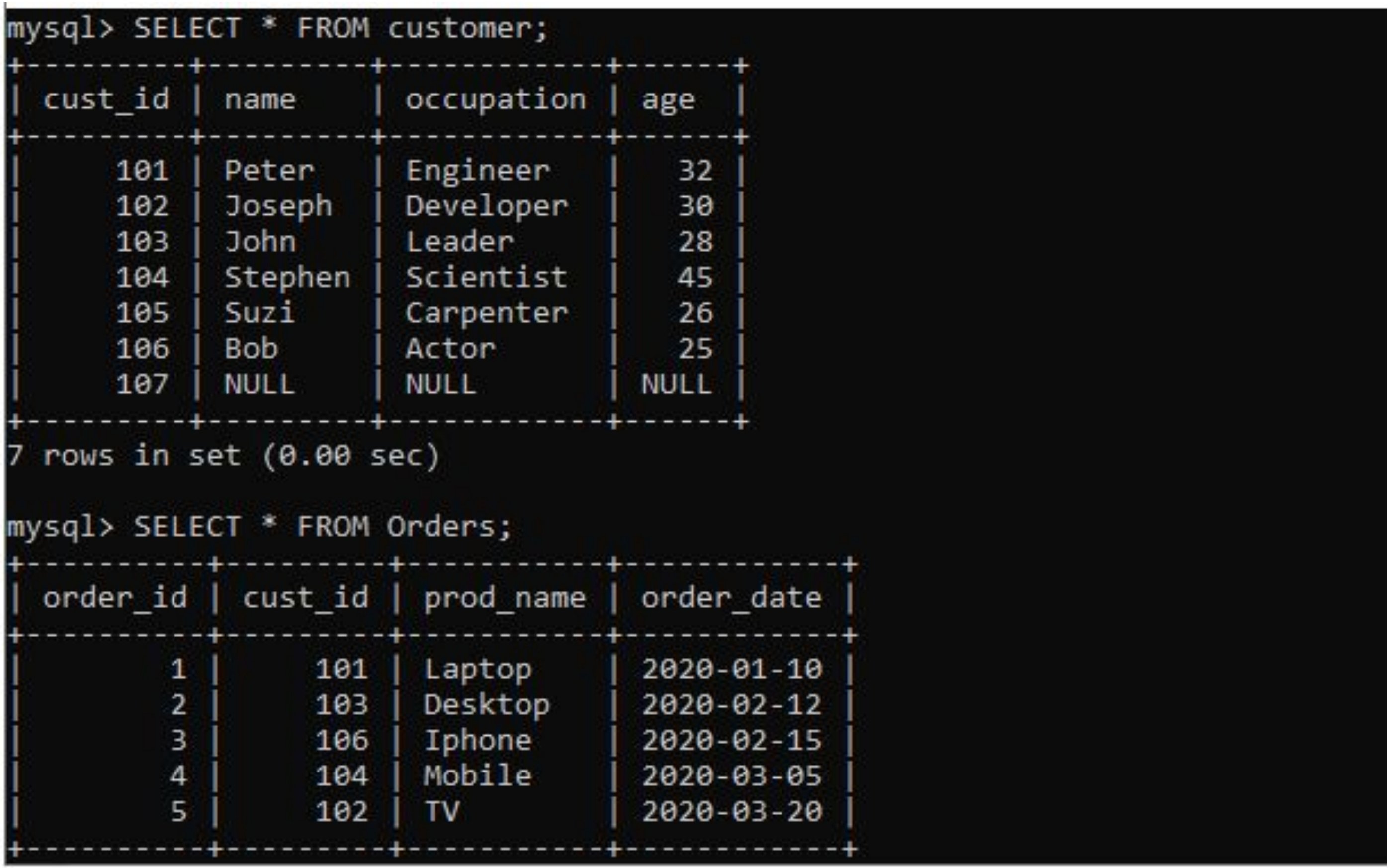
Correlated Subqueries

EXISTS and NOT EXISTS

Correlated Subqueries

The name of correlated subqueries means that a subquery is correlated with the outer query. The correlation comes from the fact that the subquery uses information from the outer query and the subquery executes once for every row in the outer query.



```
mysql> SELECT * FROM customer;
+---------+---------+------------+------+
| cust_id | name    | occupation | age  |
+---------+---------+------------+------+
|     101 | Peter   | Engineer   |   32 |
|     102 | Joseph  | Developer  |   30 |
|     103 | John    | Leader     |   28 |
|     104 | Stephen | Scientist  |   45 |
|     105 | Suzi    | Carpenter  |   26 |
|     106 | Bob     | Actor      |   25 |
|     107 | NULL    | NULL       | NULL |
+---------+---------+------------+------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Orders;
+----------+---------+-----------+------------+
| order_id | cust_id | prod_name | order_date |
+----------+---------+-----------+------------+
|        1 |     101 | Laptop    | 2020-01-10 |
|        2 |     103 | Desktop   | 2020-02-12 |
|        3 |     106 | Iphone    | 2020-02-15 |
|        4 |     104 | Mobile    | 2020-03-05 |
|        5 |     102 | TV        | 2020-03-20 |
+----------+---------+-----------+------------+
```

EXISTS

The EXISTS operator in MySQL is a type of Boolean operator which returns the **true or false** result. It is used in combination with a subquery and checks the existence of data in a subquery.

It means if a subquery returns any record, this operator returns true. Otherwise, it will return false. The true value is always represented numeric value 1, and the false value represents 0. We can use it with SELECT, UPDATE, DELETE, INSERT statement.

Syntax:

```
SELECT col_names
 FROM tab_name
 WHERE [NOT] EXISTS (
     SELECT col_names
     FROM tab_name
     WHERE condition
 );
```

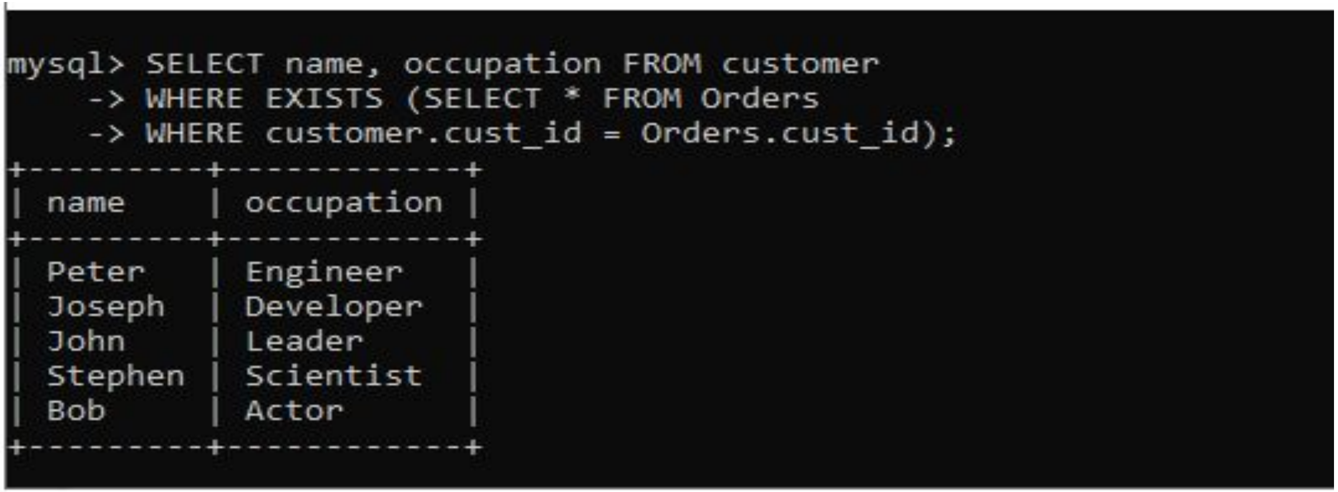
The NOT operator is used to negates the EXISTS operator. It returns true when the subquery does not return any row. Otherwise, it returns false.

In this example, we are going to use EXISTS operator to find the name and occupation of the customer who has placed at least one order:

mysql> SELECT name, occupation FROM customer

WHERE **EXISTS** (SELECT * FROM Orders

WHERE customer.cust_id = Orders.cust_id);



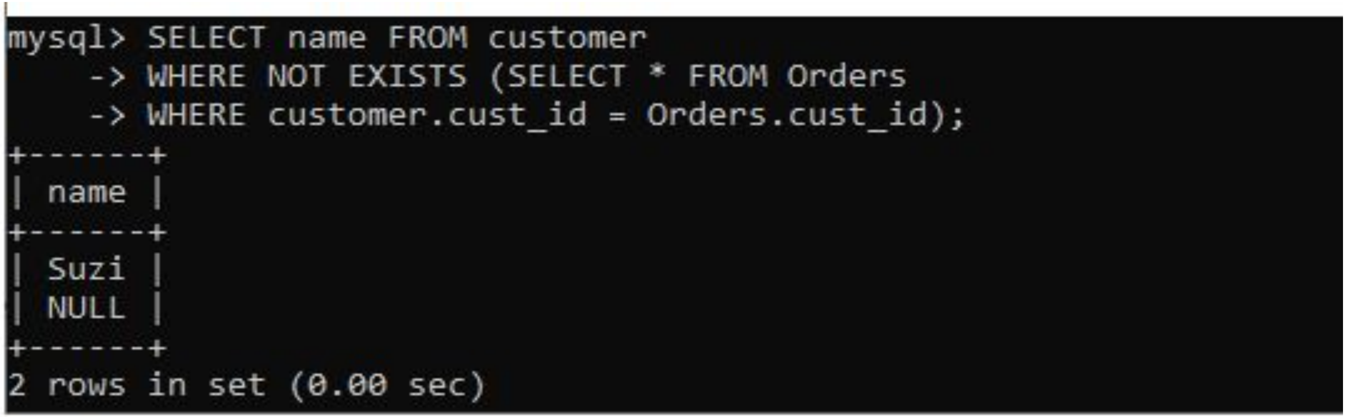
```
mysql> SELECT name, occupation FROM customer
    -> WHERE EXISTS (SELECT * FROM Orders
    -> WHERE customer.cust_id = Orders.cust_id);
+---------+------------+
| name    | occupation |
+---------+------------+
| Peter   | Engineer   |
| Joseph  | Developer  |
| John    | Leader     |
| Stephen | Scientist  |
| Bob     | Actor      |
+---------+------------+
```

Again, if we want to get the name of the customer who has not placed an order, then use the NOT EXISTS operator:

mysql> SELECT name, occupation FROM customer

WHERE NOT EXISTS (SELECT * FROM Orders

WHERE customer.cust_id = Orders.cust_id);



```
mysql> SELECT name FROM customer
    -> WHERE NOT EXISTS (SELECT * FROM Orders
    -> WHERE customer.cust_id = Orders.cust_id);
+------+
| name |
+------+
| Suzi |
| NULL |
+------+
2 rows in set (0.00 sec)
```