586. Customer Placing the Largest Number of Orders

June 9, 2017 | 23.8K views

Query the **customer_number** from the ***orders*** table for the customer who has placed the largest number of orders.

It is guaranteed that exactly one customer will have placed more orders than any other customer.

The ***orders*** table is defined as follows:

```
| Column            | Type      |
|------------------|-----------|
| order_number (PK) | int       |
| customer_number   | int       |
| order_date        | date      |
| required_date     | date      |
| shipped_date      | date      |
| status            | char(15)  |
| comment           | char(200) |
```

Sample Input

```
| order_number | customer_number | order_date | required_date | shipped_date | status |
|-------------|----------------|-----------|--------------|-------------|--------
| 1           | 1              | 2017-04-09 | 2017-04-13   | 2017-04-12  | Closed
| 2           | 2              | 2017-04-15 | 2017-04-20   | 2017-04-18  | Closed
| 3           | 3              | 2017-04-16 | 2017-04-25   | 2017-04-20  | Closed
| 4           | 3              | 2017-04-18 | 2017-04-28   | 2017-04-25  | Closed
```

Sample Output

```
| customer_number |
|----------------|
| 3              |
```

Explanation

```
The customer with number '3' has two orders, which is greater than either customer '1'
So the result is customer_number '3'.
```

Follow up: *What if more than one customer have the largest number of orders, can you find all the customer_number in this case?*

Solution

Approach: Using `LIMIT` [Accepted]

Algorithm

First, we can select the **customer_number** and the according count of orders using `GROUP BY`.

```
SELECT
    customer_number, COUNT(*)
FROM
    orders
GROUP BY customer_number
```

customer_number	COUNT(*)
1	1
2	1
3	2

Then, the **customer_number** of first record is the result after sorting them by order count descending.

customer_number	COUNT(*)
3	2

In MySQL, the LIMIT clause can be used to constrain the number of rows returned by the SELECT statement. It takes one or two nonnegative numeric arguments, the first of which specifies the offset of the first row to return, and the second specifies the maximum number of rows to return. The offset of the initial row is 0 (not 1).

It can be used with only one argument, which specifies the number of rows to return from the beginning of the result set. So `LIMIT 1` will return the first record.

MySQL

```
SELECT
    customer_number
FROM
    orders
GROUP BY customer_number
ORDER BY COUNT(*) DESC
LIMIT 1
;
```