Easy

Table: Orders

```
+-----------------+----------+
| Column Name     | Type     |
+-----------------+----------+
| order_number    | int      |
| customer_number | int      |
+-----------------+----------+
```
order_number is the primary key (column with unique values) for this table.
This table contains information about the order ID and the customer ID.

Write a solution to find the customer_number for the customer who has placed **the largest number of orders**.

The test cases are generated so that **exactly one customer** will have placed more orders than any other customer.

The result format is in the following example.

Example 1:

Input:
Orders table:
```
+--------------+-----------------+
| order_number | customer_number |
+--------------+-----------------+
| 1            | 1               |
| 2            | 2               |
| 3            | 3               |
| 4            | 3               |
+--------------+-----------------+
```
Output:
```
+-----------------+
| customer_number |
+-----------------+
| 3               |
+-----------------+
```
Explanation:
The customer with number 3 has two orders, which is greater than either customer 1 or 2 because each of them only has one order.
So the result is customer_number 3.

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

```
# Write your MySQL query statement below
SELECT customer_number
FROM Orders
GROUP BY customer_number
ORDER BY COUNT(order_number) DESC
LIMIT 1
```