**Select**

1. Table: Customer

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| id | int |

| name | varchar |

| referee\_id | int |

+-------------+---------+

id is the primary key column for this table.

Each row of this table indicates the id of a customer, their name, and the id of the customer who referred them.

Write an SQL query to report the IDs of the customer that are **not referred by** the customer with id = 2.

Return the result table in **any order**.

The query result format is in the following example.

**Example 1:**

**Input:**

Customer table:

+----+------+------------+

| id | name | referee\_id |

+----+------+------------+

| 1 | Will | null |

| 2 | Jane | null |

| 3 | Alex | 2 |

| 4 | Bill | null |

| 5 | Zack | 1 |

| 6 | Mark | 2 |

+----+------+------------+

**Output:**

+------+

| name |

+------+

| Will |

| Jane |

| Bill |

| Zack |

+------+

Answer:

select name from Customer

where referee\_id !=2 OR referee\_id IS NULL;

referee\_id IS NULL; is very important because MySQL uses three-valued logic -- TRUE, FALSE and UNKNOWN. Anything compared to NULL evaluates to the third value: UNKNOWN. That “anything” includes NULL itself! That’s why MySQL provides the IS NULL and IS NOT NULL operators to specifically check for NULL. Thus, one more condition 'referee\_id IS NULL' should be added to the WHERE clause as below.

Table: Customers

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| id | int |

| name | varchar |

+-------------+---------+

id is the primary key column for this table.

Each row of this table indicates the ID and name of a customer.

Table: Orders

+-------------+------+

| Column Name | Type |

+-------------+------+

| id | int |

| customerId | int |

+-------------+------+

id is the primary key column for this table.

customerId is a foreign key of the ID from the Customers table.

Each row of this table indicates the ID of an order and the ID of the customer who ordered it.

Write an SQL query to report all customers who never order anything.

Return the result table in **any order**.

The query result format is in the following example.

**Example 1:**

**Input:**

Customers table:

+----+-------+

| id | name |

+----+-------+

| 1 | Joe |

| 2 | Henry |

| 3 | Sam |

| 4 | Max |

+----+-------+

Orders table:

+----+------------+

| id | customerId |

+----+------------+

| 1 | 3 |

| 2 | 1 |

+----+------------+

**Output:**

+-----------+

| Customers |

+-----------+

| Henry |

| Max |

+-----------+

Answer:

select Customers.name AS Customers from Customers

Left join Orders on Orders.customerId =Customers.id

WHERE Orders.CustomerId IS NULL

Another solution:

select customers.name as 'Customers'

from customers

where customers.id not in

(

select customerid from orders

);

Inner Join

Only returns connected, matching rows. We might have duplicates in inner join.

Left join

Returns all connected rows, and unconnected rows from the left table. If left table value has no matching with right table value, it brings the left table values in the results and writes null for the right table columns (nulls in right)

Right join

Returns all connected rows, and unconnected rows from right table (nulls in left)

Full join

Left join+right join

Returns all connected rows and unconnected rows from both left and right tables.

Select x from y left/inner/right join y on x.id=y.custid

In above, the table in front of select statement (x) is the table that is located on the left.

**Select & Order**