

## 183. Customers Who Never Order

Easy

🔖 Topics

🔒 Companies

[SQL Schema](#) > [Pandas Schema](#) >

Table: `Customers`

+-----+-----+	
Column Name	Type
+-----+-----+	
id	int
name	varchar
+-----+-----+	

id is the primary key (column with unique values) 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 with unique values) for this table.  
customerId is a foreign key (reference columns) 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 a solution to find all customers who never order anything.

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

The 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

## MySQL:

# Write your MySQL query statement below

```
SELECT name AS Customers
FROM Customers c
LEFT JOIN Orders o
  ON c.id = o.customerId
WHERE o.id IS NULL;
```

## Pandas:

```
import pandas as pd
```

```
def find_customers(customers: pd.DataFrame, orders: pd.DataFrame) -> pd.DataFrame:
```

```
    # Left join customers with orders on id and customerId
```

```
    merged = customers.merge(orders, left_on="id", right_on="customerId", how="left")
```

```
    # Select customers where order id is null (meaning no order placed)
```

```
    result = merged[merged["customerId"].isna()][["name"]]
```

```
    # Rename column as required
```

```
    result = result.rename(columns={"name": "Customers"})
```

```
    return result
```

## PostgreSQL:

-- Write your PostgreSQL query statement below

SELECT c.name AS Customers

FROM Customers c

LEFT JOIN Orders o

ON c.id = o.customerId

WHERE o.id IS NULL;