1327. List the Products Ordered in a Period

Table: Products

```
+------------------+---------+
| Column Name      | Type    |
+------------------+---------+
| product_id       | int     |
| product_name     | varchar |
| product_category | varchar |
+------------------+---------+
```

product_id is the primary key (column with unique values) for this table.
This table contains data about the company's products.

Table: Orders

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| product_id    | int     |
| order_date    | date    |
| unit          | int     |
+---------------+---------+
```

This table may have duplicate rows.
product_id is a foreign key (reference column) to the Products table.
unit is the number of products ordered in order_date.

Write a solution to get the names of products that have at least 100 units ordered in **February 2020** and their amount.
Return the result table in **any order**.
The result format is in the following example.

Example 1:
Input:
Products table:

```
+-------------+----------------------+------------------+
| product_id  | product_name         | product_category |
+-------------+----------------------+------------------+
| 1           | Leetcode Solutions   | Book             |
| 2           | Jewels of Stringology | Book            |
| 3           | HP                   | Laptop           |
| 4           | Lenovo               | Laptop           |
| 5           | Leetcode Kit         | T-shirt          |
+-------------+----------------------+------------------+
```

Orders table:

```
+-------------+-------------+----------+
| product_id  | order_date  | unit     |
+-------------+-------------+----------+
| 1           | 2020-02-05  | 60       |
| 1           | 2020-02-10  | 70       |
| 2           | 2020-01-18  | 30       |
| 2           | 2020-02-11  | 80       |
| 3           | 2020-02-17  | 2        |
| 3           | 2020-02-24  | 3        |
| 4           | 2020-03-01  | 20       |
| 4           | 2020-03-04  | 30       |
| 4           | 2020-03-04  | 60       |
| 5           | 2020-02-25  | 50       |
| 5           | 2020-02-27  | 50       |
| 5           | 2020-03-01  | 50       |
+-------------+-------------+----------+
```

Output:

```
+-------------------+---------+
| product_name      | unit    |
+-------------------+---------+
| Leetcode Solutions | 130    |
| Leetcode Kit      | 100     |
+-------------------+---------+
```

Explanation:

Products with product_id = 1 is ordered in February a total of (60 + 70) = 130.

Products with product_id = 2 is ordered in February a total of 80.

Products with product_id = 3 is ordered in February a total of (2 + 3) = 5.

Products with product_id = 4 was not ordered in February 2020.

Products with product_id = 5 is ordered in February a total of (50 + 50) = 100.

/* Write your SQL query statement below */

```sql
select product_name, sum(unit) unit
from products p
join orders o on p.product_id = o.product_id
where o.order_date like '2020-02%'
group by product_name
having sum(unit) >= 100
```

 # Write your MySQL query statement below

```sql
with cte as(
select product_id,
```

```
        sum(unit) as co
from Orders
where order_date like '2020-02%'
group by product_id
)
select p.product_name, c.co as unit
from Products p inner join cte c on p.product_id= c.product_id
where c.co >= 100
```