[1327. List the Products Ordered in a Period](https://leetcode.com/problems/list-the-products-ordered-in-a-period/)

Easy

254

33

Companies

SQL SchemaPandas Schema

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.

INNER JOIN/JOIN is much better.

# Write your MySQL query statement below

# Select the required columns

SELECT Products.product\_name, SUM(Orders.unit) as unit

# We use a LEFT JOIN to join the two tables:

FROM Products JOIN Orders ON Products.product\_id = Orders.product\_id

# instead of using extract function. It saves more time when writing a query and it looks neat. :)

WHERE order\_date LIKE '2020-02%'

# We use the GROUP BY clause to find the sum of units corresponding to product\_id and calculate if sum of units >= 100

group by Products.product\_name

# HAVING unit >= 100

Having sum(Orders.unit) >=100