Problem 1549: The Most Recent Orders for Each Product

Problem Information

Difficulty: Medium
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Table:

Customers

+---------------+---------+ | Column Name | Type | +---------------+---------+ | customer_id | int | | name | varchar | +---------------+---------+ customer_id is the column with unique values for this table. This table contains information about the customers.

Table:

Orders

+---------------+---------+ | Column Name | Type | +---------------+---------+ | order_id | int | | order_date | date | | customer_id | int | | product_id | int | +---------------+---------+ order_id is the column with unique values for this table. This table contains information about the orders made by customer_id. There will be no product ordered by the same user

more than once

in one day.

Table:

Products

+---------------+---------+ | Column Name | Type | +---------------+---------+ | product_id | int | | product_name | varchar | | price | int | +---------------+---------+ product_id is the column with unique values for this table. This table contains information about the Products.

Write a solution to find the most recent order(s) of each product.

Return the result table ordered by

product_name

in ascending order and in case of a tie by the

product_id

in

ascending order

. If there still a tie, order them by

order_id

in

ascending order

.

The result format is in the following example.

Example 1:

Input:

Customers table: +-------------+-----------+ | customer_id | name | +-------------+-----------+ | 1 | Winston | | 2 | Jonathan | | 3 | Annabelle | | 4 | Marwan | | 5 | Khaled | +-------------+-----------+ Orders table: +----------+------------+-------------+------------+ | order_id | order_date | customer_id | product_id | +----------+------------+-------------+------------+ | 1 | 2020-07-31 | 1 | 1 | | 2 | 2020-07-30 | 2 | 2 | | 3 | 2020-08-29 | 3 | 3 | | 4 | 2020-07-29 | 4 | 1 | | 5 | 2020-06-10 | 1 | 2 | | 6

| 2020-08-01 | 2 | 1 | | 7 | 2020-08-01 | 3 | 1 | | 8 | 2020-08-03 | 1 | 2 | | 9 | 2020-08-07 | 2 | 3 | | 10 | 2020-07-15 | 1 | 2 | +----------+------------+-------------+------------+ Products table: +------------+--------------+-------+ | product_id | product_name | price | +------------+--------------+-------+ | 1 | keyboard | 120 | | 2 | mouse | 80 | | 3 | screen | 600 | | 4 | hard disk | 450 | +------------+--------------+-------+

Output:

+--------------+------------+----------+------------+ | product_name | product_id | order_id | order_date | +--------------+------------+----------+------------+ | keyboard | 1 | 6 | 2020-08-01 | | keyboard | 1 | 7 | 2020-08-01 | | mouse | 2 | 8 | 2020-08-03 | | screen | 3 | 3 | 2020-08-29 | +--------------+------------+----------+------------+

Explanation:

keyboard's most recent order is in 2020-08-01, it was ordered two times this day. mouse's most recent order is in 2020-08-03, it was ordered only once this day. screen's most recent order is in 2020-08-29, it was ordered only once this day. The hard disk was never ordered and we do not include it in the result table.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd

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

Solutions

MySQL Solution:

```
# Write your MySQL query statement below
```

MS SQL Server Solution:

```
/* Write your T-SQL query statement below */
```

PostgreSQL Solution:

```
-- Write your PostgreSQL query statement below
```

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

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