Problem 2986: Find Third Transaction

Problem Information

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

Problem Description

Table:

Transactions

+------------------+----------+ | Column Name | Type | +------------------+----------+ | user_id | int | | spend | decimal | | transaction_date | datetime | +------------------+----------+ (user_id, transaction_date) is column of unique values for this table. This table contains user_id, spend, and transaction_date.

Write a solution to find the

third transaction

(if they have at least three transactions) of every user, where the

spending

on the preceding

two transactions

is

lower

than the spending on the

third

transaction.

Return

the result table by

user_id

in

ascending

order

.

The result format is in the following example.

Example 1:

Input:

Transactions table: +---------+--------+---------------------+ | user_id | spend | transaction_date | +---------+--------+---------------------+ | 1 | 65.56 | 2023-11-18 13:49:42 | | 1 | 96.0 | 2023-11-30 02:47:26 | | 1 | 7.44 | 2023-11-02 12:15:23 | | 1 | 49.78 | 2023-11-12 00:13:46 | | 2 | 40.89 | 2023-11-21 04:39:15 | | 2 | 100.44 | 2023-11-20 07:39:34 | | 3 | 37.33 | 2023-11-03 06:22:02 | | 3 | 13.89 | 2023-11-11 16:00:14 | | 3 | 7.0 | 2023-11-29 22:32:36 | +---------+--------+---------------------+

Output

+---------+------------------------+-----------------------+ | user_id | third_transaction_spend | third_transaction_date | +---------+------------------------+-----------------------+ | 1 | 65.56 | 2023-11-18 13:49:42 | +---------+------------------------+-----------------------+

Explanation

- For user_id 1, their third transaction occurred on 2023-11-18 at 13:49:42 with an amount of $65.56, surpassing the expenditures of the previous two transactions which were $7.44 on 2023-11-02 at 12:15:23 and $49.78 on 2023-11-12 at 00:13:46. Thus, this third transaction will be included in the output table. - user_id 2 only has a total of 2 transactions, so there isn't a third transaction to consider. - For user_id 3, the amount of $7.0 for their third transaction is less than that of the preceding two transactions, so it won't be included. Output table is ordered by user_id in ascending order.

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 find_third_transaction(transactions: 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 find_third_transaction(transactions: pd.DataFrame) -> pd.DataFrame:
```