Problem 1843: Suspicious Bank Accounts

Problem Information

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

Problem Description

Table:

Accounts

+----------------+------+ | Column Name | Type | +----------------+------+ | account_id | int | | max_income | int | +----------------+------+ account_id is the column with unique values for this table. Each row contains information about the maximum monthly income for one bank account.

Table:

Transactions

+----------------+----------+ | Column Name | Type | +----------------+----------+ | transaction_id | int | | account_id | int | | type | ENUM | | amount | int | | day | datetime | +----------------+----------+ transaction_id is the column with unique values for this table. Each row contains information about one transaction. type is ENUM (category) type of ('Creditor','Debtor') where 'Creditor' means the user deposited money into their account and 'Debtor' means the user withdrew money from their account. amount is the amount of money deposited/withdrawn during the transaction.

A bank account is

suspicious

if the

total income

exceeds the

max_income

for this account for

two or more consecutive

months. The

total income

of an account in some month is the sum of all its deposits in that month (i.e., transactions of the type

'Creditor'

).

Write a solution to report the IDs of all

suspicious

bank accounts.

Return the result table

in any order

.

The result format is in the following example.

Example 1:

Input:

Accounts table: +------------+------------+ | account_id | max_income | +------------+------------+ | 3 | 21000 | | 4 | 10400 | +------------+------------+ Transactions table: +----------------+------------+----------+--------+---------------------+ | transaction_id | account_id | type | amount | day | +----------------+------------+----------+--------+---------------------+ | 2 | 3 | Creditor | 107100 | 2021-06-02 11:38:14 | | 4 | 4 | Creditor | 10400 | 2021-06-20 12:39:18 | | 11 | 4 | Debtor | 58800 | 2021-07-23 12:41:55 | | 1 | 4 | Creditor | 49300 | 2021-05-03 16:11:04 | | 15 | 3 | Debtor | 75500 | 2021-05-23 14:40:20 | | 10 | 3 | Creditor | 102100 | 2021-06-15 10:37:16 | | 14 | 4 | Creditor | 56300 | 2021-07-21 12:12:25 | | 19 | 4 | Debtor | 101100 | 2021-05-09 15:21:49 | | 8 | 3 | Creditor | 64900 | 2021-07-26 15:09:56 | | 7 | 3 | Creditor | 90900 | 2021-06-14 11:23:07 | +----------------+------------+----------+--------+---------------------+

Output:

+------------+ | account_id | +------------+ | 3 | +------------+

Explanation:

For account 3: - In 6-2021, the user had an income of 107100 + 102100 + 90900 = 300100. - In 7-2021, the user had an income of 64900. We can see that the income exceeded the max income of 21000 for two consecutive months, so we include 3 in the result table.

For account 4: - In 5-2021, the user had an income of 49300. - In 6-2021, the user had an income of 10400. - In 7-2021, the user had an income of 56300. We can see that the income exceeded the max income in May and July, but not in June. Since the account did not exceed the max income for two consecutive months, 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 suspicious_bank_accounts(accounts: pd.DataFrame, 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 suspicious_bank_accounts(accounts: pd.DataFrame, transactions:
pd.DataFrame) -> pd.DataFrame:
```