Problem 3118: Friday Purchase III

Problem Information

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

Problem Description

Table:

Purchases

+---------------+------+ | Column Name | Type | +---------------+------+ | user_id | int | | purchase_date | date | | amount_spend | int | +---------------+------+ (user_id, purchase_date, amount_spend) is the primary key (combination of columns with unique values) for this table. purchase_date will range from November 1, 2023, to November 30, 2023, inclusive of both dates. Each row contains user_id, purchase_date, and amount_spend.

Table:

Users

+-------------+------+ | Column Name | Type | +-------------+------+ | user_id | int | | membership | enum | +-------------+------+ user_id is the primary key for this table. membership is an ENUM (category) type of ('Standard', 'Premium', 'VIP'). Each row of this table indicates the user_id, membership type.

Write a solution to calculate the

total spending

by

Premium

and

VIP

members on

each Friday of every week

in November 2023. If there are

no purchases

on a

particular Friday

by

Premium

or

VIP

members, it should be considered as

0

.

Return

the result table

ordered by week of the month, and

membership

in

ascending

order

.

The result format is in the following example.

Example:

Input:

Purchases table:

+---------+---------------+--------------+ | user_id | purchase_date | amount_spend | +---------+---------------+--------------+ | 11 | 2023-11-03 | 1126 | | 15 | 2023-11-10 | 7473 | | 17 | 2023-11-17 | 2414 | | 12 | 2023-11-24 | 9692 | | 8 | 2023-11-24 | 5117 | | 1 | 2023-11-24 | 5241 | | 10 | 2023-11-22 | 8266 | | 13 | 2023-11-21 | 12000 | +---------+---------------+--------------+

Users table:

+---------+------------+ | user_id | membership | +---------+------------+ | 11 | Premium | | 15 | VIP | | 17 | Standard | | 12 | VIP | | 8 | Premium | | 1 | VIP | | 10 | Standard | | 13 | Premium | +---------+------------+

Output:

+---------------+-------------+--------------+ | week_of_month | membership | total_amount | +---------------+-------------+--------------+ | 1 | Premium | 1126 | | 1 | VIP | 0 | | 2 | Premium | 0 | | 2 | VIP | 7473 | | 3 | Premium | 0 | | 3 | VIP | 0 | | 4 | Premium | 5117 | | 4 | VIP | 14933 | +---------------+-------------+--------------+

Explanation:

During the first week of November 2023, a transaction occurred on Friday, 2023-11-03, by a Premium member amounting to $1,126. No transactions were made by VIP members on this day, resulting in a value of 0.

For the second week of November 2023, there was a transaction on Friday, 2023-11-10, and it was made by a VIP member, amounting to $7,473. Since there were no purchases by Premium members that Friday, the output shows 0 for Premium members.

Similarly, during the third week of November 2023, no transactions by Premium or VIP members occurred on Friday, 2023-11-17, which shows 0 for both categories in this week.

In the fourth week of November 2023, transactions occurred on Friday, 2023-11-24, involving one Premium member purchase of $5,117 and VIP member purchases totaling $14,933 ($9,692 from one and $5,241 from another).

Note:

The output table is ordered by week_of_month and membership 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 friday_purchases(purchases: pd.DataFrame, users: 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 friday_purchases(purchases: pd.DataFrame, users: pd.DataFrame) ->
pd.DataFrame:
```