

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.

Write a solution to calculate the **total spending** by users on **each Friday** of **every week** in **November 2023**. Output only weeks that include **at least one** purchase on a **Friday**.

Return the result table ordered by week of month in **ascending** order.

The result format is in the following example.

Example 1:

Input:
Purchases table:

	user_id	purchase_date	amount_spend
11	2023-11-07	1126	
15	2023-11-30	7473	
17	2023-11-14	2414	
12	2023-11-24	9692	
8	2023-11-03	5117	
1	2023-11-16	5241	
10	2023-11-12	8266	
13	2023-11-24	12000	

Output:

Explanation:

- During the first week of November 2023, transactions amounting to \$5,117 occurred on Friday, 2023-11-03.

- For the second week of November 2023, there were no transactions on Friday, 2023-11-10.

- Similarly, during the third week of November 2023, there were no transactions on Friday, 2023-11-17.

- In the fourth week of November 2023, two transactions took place on Friday, 2023-11-24, amounting to \$12,000 and \$9,692 respectively, summing up to a total of \$21,692.

Output table is ordered by week_of_month in ascending order.

Seen this question in a real interview before? 1/5

Accepted 3,865 / 4.8K | Acceptance Rate 80.4%

Topics

Discussion (6)

Copyright © 2025 LeetCode. All rights reserved.

● 0 Online

Code

```

1 import pandas as pd
2
3 def friday_purchases(purchases: pd.DataFrame) -> pd.DataFrame:
4
5     # Ensure datetime type
6     purchases['purchase_date'] = pd.to_datetime(purchases['purchase_date'])
7
8     # Keep only Fridays (Monday=0, ..., Friday=4)
9     friday_df = purchases[purchases['purchase_date'].dt.weekday == 4]
10
11     # Compute week of month
12     friday_df['week_of_month'] = (
13         (friday_df['purchase_date'].dt.day - 1) // 7 + 1
14     )
15
16     # Aggregate total spending per Friday
17     result = (
18         friday_df
19         .groupby(['week_of_month', 'purchase_date'], as_index=False)
20         .agg(total_amount=('amount_spent', 'sum'))
21         .sort_values('week_of_month')
22     )
23
24     return result

```

☑ Testcase | >_ Test Result

☒ Case 1

user_id	purchase_date	amount_spend
11	2023-11-07	1126
15	2023-11-30	7473
17	2023-11-14	2414