Problem List

Description | Accepted | Editorial | Solutions | Submissions

# 2066. Account Balance

Premium

Solved

Medium | Topics

SQL Schema > Pandas Schema >

Table: `Transactions`

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| account_id  | int  |
| day         | date |
| type        | ENUM |
| amount      | int  |
+-------------+------+
(account_id, day) is the primary key (combination of columns
with unique values) for this table.
Each row contains information about one transaction, including
the transaction type, the day it occurred on, and the amount.
type is an ENUM (category) of the type ('Deposit','Withdraw')
```

Write a solution to report the balance of each user after each transaction. You may assume that the balance of each account before any transaction is `0` and that the balance will never be below `0` at any moment.

Return the result table **in ascending order** by `account_id`, then by `day` in case of a tie.

The result format is in the following example.

**Example 1:**

**Input:**

Transactions table:

| account_id | day | type | amount |
|---|---|---|---|
| 1 | 2021-11-07 | Deposit | 2000 |
| 1 | 2021-11-09 | Withdraw | 1000 |
| 1 | 2021-11-11 | Deposit | 3000 |
| 2 | 2021-12-07 | Deposit | 7000 |
| 2 | 2021-12-12 | Withdraw | 7000 |

**Output:**

| account_id | day | balance |
|---|---|---|
| 1 | 2021-11-07 | 2000 |
| 1 | 2021-11-09 | 1000 |
| 1 | 2021-11-11 | 4000 |
| 2 | 2021-12-07 | 7000 |
| 2 | 2021-12-12 | 0 |

**Explanation:**

Account 1:
- Initial balance is 0.
- 2021-11-07 --> deposit 2000. Balance is 0 + 2000 = 2000.
- 2021-11-09 --> withdraw 1000. Balance is 2000 - 1000 = 1000.
- 2021-11-11 --> deposit 3000. Balance is 1000 + 3000 = 4000.

Account 2:
- Initial balance is 0.
- 2021-12-07 --> deposit 7000. Balance is 0 + 7000 = 7000.
- 2021-12-12 --> withdraw 7000. Balance is 7000 - 7000 = 0.

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

Yes | No

Accepted 12,400/15.2K | Acceptance Rate 81.8%

Topics

Discussion (7)



Submit

Code

Pandas • Auto

```python
    for index, row in transactions.iterrows():
        if row['account_id'] != account_id:
            account_id = row['account_id']
            balance = 0

        if row['type'] == 'Deposit':
            balance += row['amount']

        elif row['type'] == 'Withdraw':
            balance -= row['amount']


        transactions.loc[index, 'balance'] = balance


    return transactions[['account_id', 'day', 'balance']]
```

Saved

Ln 18, Col 40

Testcase | Test Result

Accepted Runtime: 261 ms

Case 1

Input

Transactions =

| account_id | day | type | amount |
|---|---|---|---|
| 1 | 2021-11-07 | Deposit | 2000 |
| 1 | 2021-11-09 | Withdraw | 1000 |
| 1 | 2021-11-11 | Deposit | 3000 |
| 2 | 2021-12-07 | Deposit | 7000 |
| 2 | 2021-12-12 | Withdraw | 7000 |

73 | 7 | 0 Online