Problem 1321: Restaurant Growth

Problem Information

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

Problem Description

Table:

Customer

+---------------+---------+ | Column Name | Type | +---------------+---------+ | customer_id | int | | name | varchar | | visited_on | date | | amount | int | +---------------+---------+ In SQL,(customer_id, visited_on) is the primary key for this table. This table contains data about customer transactions in a restaurant. visited_on is the date on which the customer with ID (customer_id) has visited the restaurant. amount is the total paid by a customer.

You are the restaurant owner and you want to analyze a possible expansion (there will be at least one customer every day).

Compute the moving average of how much the customer paid in a seven days window (i.e., current day + 6 days before).

average_amount

should be

rounded to two decimal places

.

Return the result table ordered by

visited_on

in ascending order

.

The result format is in the following example.

Example 1:

Input:

Customer table: +-------------+--------------+--------------+-------------+ | customer_id | name | visited_on | amount | +-------------+--------------+--------------+-------------+ | 1 | Jhon | 2019-01-01 | 100 | | 2 | Daniel | 2019-01-02 | 110 | | 3 | Jade | 2019-01-03 | 120 | | 4 | Khaled | 2019-01-04 | 130 | | 5 | Winston | 2019-01-05 | 110 | | 6 | Elvis | 2019-01-06 | 140 | | 7 | Anna | 2019-01-07 | 150 | | 8 | Maria | 2019-01-08 | 80 | | 9 | Jaze | 2019-01-09 | 110 | | 1 | Jhon | 2019-01-10 | 130 | | 3 | Jade | 2019-01-10 | 150 | +-------------+--------------+--------------+-------------+

Output:

+--------------+--------------+----------------+ | visited_on | amount | average_amount | +--------------+--------------+----------------+ | 2019-01-07 | 860 | 122.86 | | 2019-01-08 | 840 | 120 | | 2019-01-09 | 840 | 120 | | 2019-01-10 | 1000 | 142.86 | +--------------+--------------+----------------+

Explanation:

1st moving average from 2019-01-01 to 2019-01-07 has an average_amount of (100 + 110 + 120 + 130 + 110 + 140 + 150)/7 = 122.86 2nd moving average from 2019-01-02 to 2019-01-08 has an average_amount of (110 + 120 + 130 + 110 + 140 + 150 + 80)/7 = 120 3rd moving average from 2019-01-03 to 2019-01-09 has an average_amount of (120 + 130 + 110 + 140 + 150 + 80 + 110)/7 = 120 4th moving average from 2019-01-04 to 2019-01-10 has an average_amount of (130 + 110 + 140 + 150 + 80 + 110 + 130 + 150)/7 = 142.86

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 restaurant_growth(customer: 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 restaurant_growth(customer: pd.DataFrame) -> pd.DataFrame:
```