Problem 1193: Monthly Transactions I

Problem Information

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

Problem Description

Table:

Transactions

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | country | varchar | | state | enum | | amount | int | | trans_date | date | +---------------+---------+ id is the primary key of this table. The table has information about incoming transactions. The state column is an enum of type ["approved", "declined"].

Write an SQL query to find for each month and country, the number of transactions and their total amount, the number of approved transactions and their total amount.

Return the result table in

any order

.

The query result format is in the following example.

Example 1:

Input:

Transactions table: +------+---------+----------+--------+------------+ | id | country | state | amount | trans_date | +------+---------+----------+--------+------------+ | 121 | US | approved | 1000 | 2018-12-18 | | 122 | US | declined | 2000 | 2018-12-19 | | 123 | US | approved | 2000 |

2019-01-01 | | 124 | DE | approved | 2000 | 2019-01-07 |
+------+---------+----------+--------+------------+

Output:

+----------+---------+-------------+---------------+-------------------+----------------------+ | month | country | trans_count | approved_count | trans_total_amount | approved_total_amount | +----------+---------+-------------+---------------+-------------------+----------------------+ | 2018-12 | US | 2 | 1 | 3000 | 1000 | | 2019-01 | US | 1 | 1 | 2000 | 2000 | | 2019-01 | DE | 1 | 1 | 2000 | 2000 | +----------+---------+-------------+---------------+-------------------+----------------------+

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 monthly_transactions(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 monthly_transactions(transactions: pd.DataFrame) -> pd.DataFrame:
```