Problem 2985: Calculate Compressed Mean

Problem Information

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

Problem Description

Table:

Orders

+-------------------+------+ | Column Name | Type | +-------------------+------+ | order_id | int | | item_count | int | | order_occurrences | int | +-------------------+------+ order_id is column of unique values for this table. This table contains order_id, item_count, and order_occurrences.

Write a solution to calculate the

average

number of items per order, rounded to

2

decimal places

.

Return

the result table

in

any

order

.

The result format is in the following example.

Example 1:

Input:

Orders table: +----------+------------+-------------------+ | order_id | item_count | order_occurrences | +----------+------------+-------------------+ | 10 | 1 | 500 | | 11 | 2 | 1000 | | 12 | 3 | 800 | | 13 | 4 | 1000 | +----------+------------+-------------------+

Output

+-------------------------+ | average_items_per_order | +-------------------------+ | 2.70 | +-------------------------+

Explanation

The calculation is as follows: - Total items: (1 * 500) + (2 * 1000) + (3 * 800) + (4 * 1000) = 8900 - Total orders: 500 + 1000 + 800 + 1000 = 3300 - Therefore, the average items per order is 8900 / 3300 = 2.70

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 compressed_mean(orders: 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 compressed_mean(orders: pd.DataFrame) -> pd.DataFrame:
```