Problem 2893: Calculate Orders Within Each Interval

Problem Information

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

Problem Description

Table:

Orders

+-------------+------+ | Column Name | Type | +-------------+------+ | minute | int | | order_count | int | +-------------+------+ minute is the primary key for this table. Each row of this table contains the minute and number of orders received during that specific minute. The total number of rows will be a multiple of 6.

Write a query to calculate

total

orders

within each

interval

. Each interval is defined as a combination of

6

minutes.

Minutes

1

to

6

fall within interval

1

, while minutes

7

to

12

belong to interval

2

, and so forth.

Return

the result table ordered by

interval_no

in

ascending

order.

The result format is in the following example.

Example 1:

Input:

Orders table: +--------+-------------+ | minute | order_count | +--------+-------------+ | 1 | 0 | | 2 | 2 | | 3 | 4 | | 4 | 6 | | 5 | 1 | | 6 | 4 | | 7 | 1 | | 8 | 2 | | 9 | 4 | | 10 | 1 | | 11 | 4 | | 12 | 6 | +--------+-------------+

Output:

+-------------+--------------+ | interval_no | total_orders | +-------------+--------------+ | 1 | 17 | | 2 | 18 | +-------------+--------------+

Explanation:

- Interval number 1 comprises minutes from 1 to 6. The total orders in these six minutes are (0 + 2 + 4 + 6 + 1 + 4) = 17. - Interval number 2 comprises minutes from 7 to 12. The total orders in these six minutes are (1 + 2 + 4 + 1 + 4 + 6) = 18. Returning table orderd by interval_no in ascending order.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd

def calculate_runs(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 */
```

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

def calculate_runs(orders: pd.DataFrame) -> pd.DataFrame:
```