

Problem 1651: Hopper Company Queries III

Problem Information

Difficulty: Hard

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Table:

Drivers

+-----+-----+ | Column Name | Type | +-----+-----+ | driver_id | int | | join_date | date | +-----+-----+ driver_id is the column with unique values for this table. Each row of this table contains the driver's ID and the date they joined the Hopper company.

Table:

Rides

+-----+-----+ | Column Name | Type | +-----+-----+ | ride_id | int | | user_id | int | | requested_at | date | +-----+-----+ ride_id is the column with unique values for this table. Each row of this table contains the ID of a ride, the user's ID that requested it, and the day they requested it. There may be some ride requests in this table that were not accepted.

Table:

AcceptedRides

+-----+-----+ | Column Name | Type | +-----+-----+ | ride_id | int | | driver_id | int | | ride_distance | int | | ride_duration | int | +-----+-----+ ride_id is the column with unique values for this table. Each row of this table contains some information about an accepted ride. It is guaranteed that each accepted ride exists in the Rides table.

Write a solution to compute the

average_ride_distance

and

average_ride_duration

of every 3-month window starting from

January - March 2020

to

October - December 2020

. Round

average_ride_distance

and

average_ride_duration

to the nearest

two decimal places

The

average_ride_distance

is calculated by summing up the total

ride_distance

values from the three months and dividing it by

3

. The

average_ride_duration

is calculated in a similar way.

Return the result table ordered by

month

in ascending order, where

month

is the starting month's number (January is

1

, February is

2

, etc.).

The result format is in the following example.

Example 1:

Input:

Drivers table: +-----+-----+ | driver_id | join_date | -----+-----+ | 10 |
2019-12-10 || 8 | 2020-1-13 || 5 | 2020-2-16 || 7 | 2020-3-8 || 4 | 2020-5-17 || 1 | 2020-10-24
|| 6 | 2021-1-5 | -----+-----+ Rides table: +-----+-----+-----+ | ride_id |
user_id | requested_at | -----+-----+-----+ | 6 | 75 | 2019-12-9 || 1 | 54 | 2020-2-9 |
| 10 | 63 | 2020-3-4 || 19 | 39 | 2020-4-6 || 3 | 41 | 2020-6-3 || 13 | 52 | 2020-6-22 || 7 | 69 |
2020-7-16 || 17 | 70 | 2020-8-25 || 20 | 81 | 2020-11-2 || 5 | 57 | 2020-11-9 || 2 | 42 |
2020-12-9 || 11 | 68 | 2021-1-11 || 15 | 32 | 2021-1-17 || 12 | 11 | 2021-1-19 || 14 | 18 |

2021-1-27 | +-----+-----+-----+ AcceptedRides table:

	ride_id	driver_id	ride_distance	ride_duration
+-----+-----+-----+ 10 10 63 38	13	10	73	96
100 28 17 7 119 68	20	1	121	92
5 7 42 101	2	4	6	38
11 8 37 43	15	8	108	82
12 8 38 34	14	1	90	74
+-----+-----+-----+				

Output:

	month	average_ride_distance	average_ride_duration
+-----+-----+-----+ 1 21.00 12.67	2	21.00	12.67
21.00 12.67 3 21.00 12.67	4	24.33	32.00
4 5 57.67 41.33	6	97.33	64.00
7 73.00 32.00 8 39.67 22.67	9	54.33	64.33
10 56.33 77.00 +-----+-----+			

Explanation:

By the end of January --> average_ride_distance = $(0+0+63)/3=21$, average_ride_duration = $(0+0+38)/3=12.67$ By the end of February --> average_ride_distance = $(0+63+0)/3=21$, average_ride_duration = $(0+38+0)/3=12.67$ By the end of March --> average_ride_distance = $(63+0+0)/3=21$, average_ride_duration = $(38+0+0)/3=12.67$ By the end of April --> average_ride_distance = $(0+0+73)/3=24.33$, average_ride_duration = $(0+0+96)/3=32.00$ By the end of May --> average_ride_distance = $(0+73+100)/3=57.67$, average_ride_duration = $(0+96+28)/3=41.33$ By the end of June --> average_ride_distance = $(73+100+119)/3=97.33$, average_ride_duration = $(96+28+68)/3=64.00$ By the end of July --> average_ride_distance = $(100+119+0)/3=73.00$, average_ride_duration = $(28+68+0)/3=32.00$ By the end of August --> average_ride_distance = $(119+0+0)/3=39.67$, average_ride_duration = $(68+0+0)/3=22.67$ By the end of September --> average_ride_distance = $(0+0+163)/3=54.33$, average_ride_duration = $(0+0+193)/3=64.33$ By the end of October --> average_ride_distance = $(0+163+6)/3=56.33$, average_ride_duration = $(0+193+38)/3=77.00$

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 hopper_company_queries(drivers: pd.DataFrame, rides: pd.DataFrame,
accepted_rides: 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 hopper_company_queries(drivers: pd.DataFrame, rides: pd.DataFrame,
accepted_rides: pd.DataFrame) -> pd.DataFrame:
```