Problem 585: Investments in 2016

Problem Information

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

Problem Description

Table:

Insurance

+-------------+-------+ | Column Name | Type | +-------------+-------+ | pid | int | | tiv_2015 | float | | tiv_2016 | float | | lat | float | | lon | float | +-------------+-------+ pid is the primary key (column with unique values) for this table. Each row of this table contains information about one policy where: pid is the policyholder's policy ID. tiv_2015 is the total investment value in 2015 and tiv_2016 is the total investment value in 2016. lat is the latitude of the policy holder's city. It's guaranteed that lat is not NULL. lon is the longitude of the policy holder's city. It's guaranteed that lon is not NULL.

Write a solution to report the sum of all total investment values in 2016

tiv_2016

, for all policyholders who:

have the same

tiv_2015

value as one or more other policyholders, and

are not located in the same city as any other policyholder (i.e., the (

lat, lon

) attribute pairs must be unique).

Round

tiv_2016

to

two decimal places

.

The result format is in the following example.

Example 1:

Input:

Insurance table: +-----+----------+----------+-----+-----+ | pid | tiv_2015 | tiv_2016 | lat | lon | +-----+----------+----------+-----+-----+ | 1 | 10 | 5 | 10 | 10 | | 2 | 20 | 20 | 20 | 20 | | 3 | 10 | 30 | 20 | 20 | | 4 | 10 | 40 | 40 | 40 | +-----+----------+----------+-----+-----+

Output:

+----------+ | tiv_2016 | +----------+ | 45.00 | +----------+

Explanation:

The first record in the table, like the last record, meets both of the two criteria. The tiv_2015 value 10 is the same as the third and fourth records, and its location is unique.

The second record does not meet any of the two criteria. Its tiv_2015 is not like any other policyholders and its location is the same as the third record, which makes the third record fail, too. So, the result is the sum of tiv_2016 of the first and last record, which is 45.

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 find_investments(insurance: 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 find_investments(insurance: pd.DataFrame) -> pd.DataFrame:
```