Problem 197: Rising Temperature

Problem Information

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

Problem Description

Table:

Weather

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | recordDate | date | | temperature | int | +---------------+---------+ id is the column with unique values for this table. There are no different rows with the same recordDate. This table contains information about the temperature on a certain day.

Write a solution to find all dates'

id

with higher temperatures compared to its previous dates (yesterday).

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Weather table: +----+------------+-------------+ | id | recordDate | temperature | +----+------------+-------------+ | 1 | 2015-01-01 | 10 | | 2 | 2015-01-02 | 25 | | 3 | 2015-01-03 | 20 | | 4 | 2015-01-04 | 30 | +----+------------+-------------+

Output:

+----+ | id | +----+ | 2 | | 4 | +----+

Explanation:

In 2015-01-02, the temperature was higher than the previous day (10 -> 25). In 2015-01-04, the temperature was higher than the previous day (20 -> 30).

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 rising_temperature(weather: 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 rising_temperature(weather: pd.DataFrame) -> pd.DataFrame:
```