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).

SOLUTION

MySQL:

```
SELECT w2.id
FROM Weather w1
CROSS JOIN Weather w2
WHERE DATEDIFF(w2.recordDate, w1.recordDate) = 1 AND w2.temperature > w1.temperature;
```

PostgreSQL:

```
WITH w AS(
    SELECT *
    FROM Weather
    ORDER BY recordDate),
t AS(
    SELECT CASE WHEN w2.temperature > w1.temperature THEN w2.id
    END AS id
    FROM w w1
    JOIN w w2
    ON w2.recordDate - w1.recordDate = 1)
SELECT Id
FROM t
WHERE t IS NOT NULL;
```