

Problem 601: Human Traffic of Stadium

Problem Information

Difficulty: **Hard**

Acceptance Rate: 50.18%

Paid Only: No

Tags: Database

Problem Description

Table: `Stadium`

+-----+-----+ | Column Name | Type | +-----+-----+ | id | int | | visit_date | date | | people | int | +-----+-----+ visit_date is the column with unique values for this table. Each row of this table contains the visit date and visit id to the stadium with the number of people during the visit. As the id increases, the date increases as well.

Write a solution to display the records with three or more rows with **consecutive** `id`'s, and the number of people is greater than or equal to 100 for each.

Return the result table ordered by `visit_date` in **ascending order**.

The result format is in the following example.

Example 1:

Input: Stadium table: +-----+-----+-----+ | id | visit_date | people |
+-----+-----+-----+ | 1 | 2017-01-01 | 10 | | 2 | 2017-01-02 | 109 | | 3 | 2017-01-03 | 150 | | 4 | 2017-01-04 | 99 | | 5 | 2017-01-05 | 145 | | 6 | 2017-01-06 | 1455 | | 7 | 2017-01-07 | 199 | | 8 | 2017-01-09 | 188 | +-----+-----+
Output:
+-----+-----+-----+ | id | visit_date | people | +-----+-----+ | 5 | 2017-01-05 | 145 | | 6 | 2017-01-06 | 1455 | | 7 | 2017-01-07 | 199 | | 8 | 2017-01-09 | 188 | +-----+-----+
Explanation: The four rows with ids 5, 6, 7, and 8 have consecutive ids and each of them has ≥ 100 people attended. Note that row 8 was included even though the visit_date was not the next day after row 7. The rows with ids 2 and 3 are not included because we need at least three consecutive ids.

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
```