Problem 1867: Orders With Maximum Quantity Above Average

Problem Information

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

Problem Description

Table:

OrdersDetails

+-------------+------+ | Column Name | Type | +-------------+------+ | order_id | int | | product_id | int | | quantity | int | +-------------+------+ (order_id, product_id) is the primary key (combination of columns with unique values) for this table. A single order is represented as multiple rows, one row for each product in the order. Each row of this table contains the quantity ordered of the product product_id in the order order_id.

You are running an e-commerce site that is looking for

imbalanced orders

. An

imbalanced order

is one whose

maximum

quantity is

strictly greater

than the

average

quantity of

every order (including itself)

.

The

average

quantity of an order is calculated as

(total quantity of all products in the order) / (number of different products in the order)

. The

maximum

quantity of an order is the highest

quantity

of any single product in the order.

Write a solution to find the

order_id

of all

imbalanced orders

.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

OrdersDetails table: +----------+------------+----------+ | order_id | product_id | quantity | +----------+------------+----------+ | 1 | 1 | 12 | | 1 | 2 | 10 | | 1 | 3 | 15 | | 2 | 1 | 8 | | 2 | 4 | 4 | | 2 | 5 | 6 | | 3 | 3 | 5 | | 3 | 4 | 18 | | 4 | 5 | 2 | | 4 | 6 | 8 | | 5 | 7 | 9 | | 5 | 8 | 9 | | 3 | 9 | 20 | | 2 | 9 | 4 | +----------+------------+----------+

Output:

+----------+ | order_id | +----------+ | 1 | | 3 | +----------+

Explanation:

The average quantity of each order is: - order_id=1: (12+10+15)/3 = 12.3333333 - order_id=2: (8+4+6+4)/4 = 5.5 - order_id=3: (5+18+20)/3 = 14.333333 - order_id=4: (2+8)/2 = 5 - order_id=5: (9+9)/2 = 9

The maximum quantity of each order is: - order_id=1: max(12, 10, 15) = 15 - order_id=2: max(8, 4, 6, 4) = 8 - order_id=3: max(5, 18, 20) = 20 - order_id=4: max(2, 8) = 8 - order_id=5: max(9, 9) = 9

Orders 1 and 3 are imbalanced because they have a maximum quantity that exceeds the average quantity of every order.

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 orders_above_average(orders_details: 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 orders_above_average(orders_details: pd.DataFrame) -> pd.DataFrame:
```