Problem 3053: Classifying Triangles by Lengths

Problem Information

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

Problem Description

Table:

Triangles

+-------------+------+ | Column Name | Type | +-------------+------+ | A | int | | B | int | | C | int | +-------------+------+ (A, B, C) is the primary key for this table. Each row include the lengths of each of a triangle's three sides.

Write a query to find the type of

triangle

. Output one of the following for each row:

Equilateral

: It's a triangle with

3

sides of equal length.

Isosceles

: It's a triangle with

2

sides of equal length.

Scalene

: It's a triangle with

3

sides of differing lengths.

Not A Triangle:

The given values of

A

,

B

, and

C

don't form a triangle.

Return

the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Triangles table: +----+----+----+ | A | B | C | +----+----+----+ | 20 | 20 | 23 | | 20 | 20 | 20 | | 20 | 21 | 22 | | 13 | 14 | 30 | +----+----+----+

Output:

+----------------+ | triangle_type | +----------------+ | Isosceles | | Equilateral | | Scalene | | Not A Triangle | +----------------+

Explanation:

- Values in the first row from an Isosceles triangle, because A = B. - Values in the second row from an Equilateral triangle, because A = B = C. - Values in the third row from an Scalene triangle, because A != B != C. - Values in the fourth row cannot form a triangle, because the combined value of sides A and B is not larger than that of side C.

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 type_of_triangle(triangles: 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 type_of_triangle(triangles: pd.DataFrame) -> pd.DataFrame:
```