Problem 1440: Evaluate Boolean Expression

Problem Information

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

Problem Description

Table

Variables

:

+---------------+---------+ | Column Name | Type | +---------------+---------+ | name | varchar | | value | int | +---------------+---------+ In SQL, name is the primary key for this table. This table contains the stored variables and their values.

Table

Expressions

:

+---------------+---------+ | Column Name | Type | +---------------+---------+ | left_operand | varchar | | operator | enum | | right_operand | varchar | +---------------+---------+ In SQL, (left_operand, operator, right_operand) is the primary key for this table. This table contains a boolean expression that should be evaluated. operator is an enum that takes one of the values ('<', '>', '=') The values of left_operand and right_operand are guaranteed to be in the Variables table.

Evaluate the boolean expressions in

Expressions

table.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Variables table: +------+-------+ | name | value | +------+-------+ | x | 66 | | y | 77 | +------+-------+ Expressions table: +--------------+----------+---------------+ | left_operand | operator | right_operand | +--------------+----------+---------------+ | x | > | y | | x | < | y | | x | = | y | | y | > | x | | y | < | x | | x | = | x | +--------------+----------+---------------+

Output:

+--------------+----------+---------------+-------+ | left_operand | operator | right_operand | value | +--------------+----------+---------------+-------+ | x | > | y | false | | x | < | y | true | | x | = | y | false | | y | > | x | true | | y | < | x | false | | x | = | x | true | +--------------+----------+---------------+-------+

Explanation:

As shown, you need to find the value of each boolean expression in the table using the variables table.

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 eval_expression(variables: pd.DataFrame, expressions: 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 eval_expression(variables: pd.DataFrame, expressions: pd.DataFrame) ->
pd.DataFrame:
```