4. Evaluate Boolean Expression

SQL Schema

Table Variables:

+---------------+---------+

| Column Name | Type |

+---------------+---------+

| name | varchar |

| value | int |

+---------------+---------+

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 |

+---------------+---------+

(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.

Write an SQL query to evaluate the boolean expressions in Expressions table.

Return the result table in any order.

The query result format is in the following example.

Programm:

```
import sqlite3
conn=sqlite3.connect(':memory:')
cursor=conn.cursor()
cursor.execute('''
CREATE TABLE Variables (
   name TEXT PRIMARY KEY,
   value INTEGER
)
''')
variables_data=[('A', 1),
                ('B', 2),
                 ('C', 3)]
cursor.executemany('INSERT INTO Variables (name, value) VALUES (?, ?)', variables_data)
cursor.execute('''
CREATE TABLE Expressions (
   left_operand TEXT,
   operator TEXT,
   right_operand TEXT,
   PRIMARY KEY (left_operand, operator, right_operand)
)
''')
expressions_data = [('A', '<', 'B'),
                    ('B', '>', 'C'),
                    ('A', '=', 'A')]
cursor.executemany('INSERT INTO Expressions (left_operand, operator, right_operand) VALUES
(?, ?, ?)', expressions_data)
query = '''
SELECT
   e.left_operand,
   e.operator,
   e.right_operand,
```

```
    CASE
        WHEN e.operator = '<' THEN v1.value < v2.value
        WHEN e.operator = '>' THEN v1.value > v2.value
        WHEN e.operator = '=' THEN v1.value = v2.value
    END AS result
FROM
    Expressions e
JOIN
    Variables v1 ON e.left_operand = v1.name
JOIN
    Variables v2 ON e.right_operand = v2.name;
'''
cursor.execute(query)
results = cursor.fetchall()
for row in results:
    print(row)
conn.close()
```

output:

```
S C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/daa.py
'A', '<', 'B', 1)
'B', '>', 'C', 0)
'A', '=', 'A', 1)
S C:\Users\karth>
```

Time complexity:

F(n)=o(logn)