Problem 3054: Binary Tree Nodes

Problem Information

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

Problem Description

Table:

Tree

+-------------+------+ | Column Name | Type | +-------------+------+ | N | int | | P | int | +-------------+------+ N is the column of unique values for this table. Each row includes N and P, where N represents the value of a node in Binary Tree, and P is the parent of N.

Write a solution to find the node type of the Binary Tree. Output one of the following for each node:

Root

: if the node is the root node.

Leaf

: if the node is the leaf node.

Inner

: if the node is neither root nor leaf node.

Return

the result table ordered by node value in

ascending order

.

The result format is in the following example.

Example 1:

Input:

Tree table: +---+------+ | N | P | +---+------+ | 1 | 2 | | 3 | 2 | | 6 | 8 | | 9 | 8 | | 2 | 5 | | 8 | 5 | | 5 | null | +---+------+

Output:

+---+-------+ | N | Type | +---+-------+ | 1 | Leaf | | 2 | Inner | | 3 | Leaf | | 5 | Root | | 6 | Leaf | | 8 | Inner | | 9 | Leaf | +---+-------+

Explanation:

- Node 5 is the root node since it has no parent node. - Nodes 1, 3, 6, and 9 are leaf nodes because they don't have any child nodes. - Nodes 2, and 8 are inner nodes as they serve as parents to some of the nodes in the structure.

Note:

This question is the same as

608: Tree Node.

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 binary_tree_nodes(tree: 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 binary_tree_nodes(tree: pd.DataFrame) -> pd.DataFrame:
```