05/14/2022

608. Tree Node

Table: `Tree`

```
+-------------+------+

| Column Name | Type |

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

| id          | int  |

| p_id        | int  |

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

id is the primary key column for this table.

Each row of this table contains information about the id of a node and the id of its
parent node in a tree.

The given structure is always a valid tree.
```

Each node in the tree can be one of three types:

- **"Leaf"**: if the node is a leaf node.
- **"Root"**: if the node is the root of the tree.
- **"Inner"**: If the node is neither a leaf node nor a root node.

Write an SQL query to report the type of each node in the tree.

Return the result table **ordered** by `id` **in ascending order**.

MySQL

SELECT DISTINCT t1.id,(

CASE

WHEN t1.p_id IS NULL THEN 'Root'

WHEN t1.p_id IS NOT NULL AND t2.id IS NULL THEN 'Leaf'

ELSE 'Inner'

END

) AS Type

FROM Tree t1

LEFT JOIN Tree t2

ON t1.id = t2.p_id

ORDER BY t1.id ASC

176. Second Highest Salary

Table: `Employee`

```
+-------------+------+

| Column Name | Type |

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

| id          | int  |

| salary      | int  |

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

id is the primary key column for this table.

Each row of this table contains information about the salary of an employee.
```

Write an SQL query to report the second highest salary from the `Employee` table. If there is no second highest salary, the query should report `null`.

MySQL

SELECT MAX(salary) as SecondHighestSalary

FROM Employee

WHERE salary < (SELECT MAX(salary) FROM Employee)

```
#Solution 2
SELECT (
    SELECT DISTINCT Salary
    FROM EMPLOYEE
    ORDER BY Salary DESC
    LIMIT 1 OFFSET 1
) AS SecondHighestSalary
```