Problem 1378: Replace Employee ID With The Unique Identifier

Problem Information

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

Problem Description

Table:

Employees

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | name | varchar | +---------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table contains the id and the name of an employee in a company.

Table:

EmployeeUNI

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | unique_id | int | +---------------+---------+ (id, unique_id) is the primary key (combination of columns with unique values) for this table. Each row of this table contains the id and the corresponding unique id of an employee in the company.

Write a solution to show the

unique ID

of each user, If a user does not have a unique ID replace just show

null

.

Return the result table in

any

order.

The result format is in the following example.

Example 1:

Input:

Employees table: +----+----------+ | id | name | +----+----------+ | 1 | Alice | | 7 | Bob | | 11 | Meir | | 90 | Winston | | 3 | Jonathan | +----+----------+ EmployeeUNI table: +----+-----------+ | id | unique_id | +----+-----------+ | 3 | 1 | | 11 | 2 | | 90 | 3 | +----+-----------+

Output:

+-----------+----------+ | unique_id | name | +-----------+----------+ | null | Alice | | null | Bob | | 2 | Meir | | 3 | Winston | | 1 | Jonathan | +-----------+----------+

Explanation:

Alice and Bob do not have a unique ID, We will show null instead. The unique ID of Meir is 2. The unique ID of Winston is 3. The unique ID of Jonathan is 1.

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 replace_employee_id(employees: pd.DataFrame, employee_uni: 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 replace_employee_id(employees: pd.DataFrame, employee_uni: pd.DataFrame)
-> pd.DataFrame:
```