Problem 1179: Reformat Department Table

Problem Information

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

Problem Description

Table:

Department

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | revenue | int | | month | varchar | +-------------+---------+ In SQL,(id, month) is the primary key of this table. The table has information about the revenue of each department per month. The month has values in ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"].

Reformat the table such that there is a department id column and a revenue column

for each month

.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Department table: +------+---------+-------+ | id | revenue | month | +------+---------+-------+ | 1 | 8000 | Jan | | 2 | 9000 | Jan | | 3 | 10000 | Feb | | 1 | 7000 | Feb | | 1 | 6000 | Mar | +------+---------+-------+

Output:

+------+------------+------------+------------+-----+------------+ | id | Jan_Revenue | Feb_Revenue | Mar_Revenue | ... | Dec_Revenue | +------+------------+------------+------------+-----+------------+ | 1 | 8000 | 7000 | 6000 | ... | null | | 2 | 9000 | null | null | ... | null | | 3 | null | 10000 | null | ... | null | +------+------------+------------+------------+-----+------------+

Explanation:

The revenue from Apr to Dec is null. Note that the result table has 13 columns (1 for the department id + 12 for the months).

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 reformat_table(department: 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 reformat_table(department: pd.DataFrame) -> pd.DataFrame:
```