Problem 2252: Dynamic Pivoting of a Table

Problem Information

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

Problem Description

Table:

Products

+-------------+---------+ | Column Name | Type | +-------------+---------+ | product_id | int | | store | varchar | | price | int | +-------------+---------+ (product_id, store) is the primary key (combination of columns with unique values) for this table. Each row of this table indicates the price of product_id in store. There will be at most 30 different stores in the table. price is the price of the product at this store.

Important note:

This problem targets those who have a good experience with SQL. If you are a beginner, we recommend that you skip it for now.

Implement the procedure

PivotProducts

to reorganize the

Products

table so that each row has the id of one product and its price in each store. The price should be

null

if the product is not sold in a store. The columns of the table should contain each store and they should be sorted in

lexicographical order

.

The procedure should return the table after reorganizing it.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Products table: +------------+----------+-------+ | product_id | store | price | +------------+----------+-------+ | 1 | Shop | 110 | | 1 | LC_Store | 100 | | 2 | Nozama | 200 | | 2 | Souq | 190 | | 3 | Shop | 1000 | | 3 | Souq | 1900 | +------------+----------+-------+

Output:

+------------+----------+--------+------+------+ | product_id | LC_Store | Nozama | Shop | Souq | +------------+----------+--------+------+------+ | 1 | 100 | null | 110 | null | | 2 | null | 200 | null | 190 | | 3 | null | null | 1000 | 1900 | +------------+----------+--------+------+------+

Explanation:

We have 4 stores: Shop, LC_Store, Nozama, and Souq. We first order them lexicographically to be: LC_Store, Nozama, Shop, and Souq. Now, for product 1, the price in LC_Store is 100 and in Shop is 110. For the other two stores, the product is not sold so we set the price as null. Similarly, product 2 has a price of 200 in Nozama and 190 in Souq. It is not sold in the other two stores. For product 3, the price is 1000 in Shop and 1900 in Souq. It is not sold in the other two stores.

Code Snippets

MySQL:

```
CREATE PROCEDURE PivotProducts()
BEGIN
# Write your MySQL query statement below.

END
```

MS SQL Server:

```
CREATE PROCEDURE PivotProducts AS
BEGIN
/* Write your T-SQL query statement below. */

END
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below.
CREATE OR REPLACE FUNCTION PivotProducts()
```

Oracle:

```
CREATE FUNCTION PivotProducts
RETURN SYS_REFCURSOR IS result SYS_REFCURSOR;
BEGIN
/* Write your PL/SQL query statement below */

RETURN result;
END;
```

Pandas:

```
import pandas as pd

def dynamic_pivoting_table(products: pd.DataFrame) -> pd.DataFrame:
```

Solutions

MySQL Solution:

```
CREATE PROCEDURE PivotProducts()
BEGIN
# Write your MySQL query statement below.

END
```

MS SQL Server Solution:

```
CREATE PROCEDURE PivotProducts AS
BEGIN
/* Write your T-SQL query statement below. */

END
```

PostgreSQL Solution:

```
-- Write your PostgreSQL query statement below.
CREATE OR REPLACE FUNCTION PivotProducts()
```

Oracle Solution:

```
CREATE FUNCTION PivotProducts
RETURN SYS_REFCURSOR IS result SYS_REFCURSOR;
BEGIN
/* Write your PL/SQL query statement below */

RETURN result;
END;
```

Pandas Solution:

```
import pandas as pd

def dynamic_pivoting_table(products: pd.DataFrame) -> pd.DataFrame:
```