Problem 2253: Dynamic Unpivoting 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_name

1

| int | | store_name

2

| int | | : | int | | : | int | | : | int | | store_name

n

| int | +-------------+---------+ product_id is the primary key for this table. Each row in this table indicates the product's price in n different stores. If the product is not available in a store, the price will be null in that store's column. The names of the stores may change from one testcase to another. There will be at least 1 store and at most 30 stores.

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

UnpivotProducts

to reorganize the

Products

table so that each row has the id of one product, the name of a store where it is sold, and its price in that store. If a product is not available in a store, do

not

include a row with that

product_id

and

store

combination in the result table. There should be three columns:

product_id

,

store

, and

price

.

The procedure should return the table after reorganizing it.

Return the result table in

any order

.

The query result format is in the following example.

Example 1:

Input:

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

Output:

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

Explanation:

Product 1 is sold in LC_Store and Shop with prices of 100 and 110 respectively. Product 2 is sold in Nozama and Souq with prices of 200 and 190. Product 3 is sold in Shop and Souq with prices of 1000 and 1900.

Code Snippets

MySQL:

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

END
```

MS SQL Server:

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

END
```

Oracle:

```
/* Important: For this problem, column names are case sensitive.
You should use double quotes while using the columns.
*/
CREATE FUNCTION UnpivotProducts
RETURN SYS_REFCURSOR IS result SYS_REFCURSOR;
BEGIN
/* Write your PL/SQL query statement below */

RETURN result;
END;
```

Pandas:

```
import pandas as pd

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

Solutions

MySQL Solution:

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

END
```

MS SQL Server Solution:

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

```
END
```

Oracle Solution:

```
/* Important: For this problem, column names are case sensitive.
You should use double quotes while using the columns.
*/
CREATE FUNCTION UnpivotProducts
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 find_valid_users(products: pd.DataFrame) -> pd.DataFrame:
```