Problem 1777: Product's Price for Each Store

Problem Information

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

Problem Description

Table:

Products

+-------------+---------+ | Column Name | Type | +-------------+---------+ | product_id | int | | store | enum | | price | int | +-------------+---------+ In SQL, (product_id, store) is the primary key for this table. store is a category of type ('store1', 'store2', 'store3') where each represents the store this product is available at. price is the price of the product at this store.

Find the price of each product in each store.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Products table: +-------------+--------+-------+ | product_id | store | price | +-------------+--------+-------+ | 0 | store1 | 95 | | 0 | store3 | 105 | | 0 | store2 | 100 | | 1 | store1 | 70 | | 1 | store3 | 80 | +-------------+--------+-------+

Output:

+-------------+--------+--------+--------+ | product_id | store1 | store2 | store3 | +-------------+--------+--------+--------+ | 0 | 95 | 100 | 105 | | 1 | 70 | null | 80 | +-------------+--------+--------+--------+

Explanation:

Product 0 price's are 95 for store1, 100 for store2 and, 105 for store3. Product 1 price's are 70 for store1, 80 for store3 and, it's not sold in store2.

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 products_price(products: 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 products_price(products: pd.DataFrame) -> pd.DataFrame:
```