

Problem 1421: NPV Queries

Problem Information

Difficulty: Easy

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Table:

NPV

+-----+-----+ | Column Name | Type | +-----+-----+ | id | int | | year | int | |
npv | int | +-----+-----+ (id, year) is the primary key (combination of columns with
unique values) of this table. The table has information about the id and the year of each
inventory and the corresponding net present value.

Table:

Queries

+-----+-----+ | Column Name | Type | +-----+-----+ | id | int | | year | int | |
+-----+-----+ (id, year) is the primary key (combination of columns with unique
values) of this table. The table has information about the id and the year of each inventory
query.

Write a solution to find the

npv

of each query of the

Queries

table.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

NPV table: +-----+-----+-----+ | id | year | npv | +-----+-----+-----+ | 1 | 2018 | 100 | | 7 | 2020 | 30 | | 13 | 2019 | 40 | | 1 | 2019 | 113 | | 2 | 2008 | 121 | | 3 | 2009 | 12 | | 11 | 2020 | 99 | | 7 | 2019 | 0 | +-----+-----+-----+ Queries table: +-----+-----+ | id | year | +-----+-----+ | 1 | 2019 | | 2 | 2008 | | 3 | 2009 | | 7 | 2018 | | 7 | 2019 | | 7 | 2020 | | 13 | 2019 | +-----+-----+

Output:

+-----+-----+-----+ | id | year | npv | +-----+-----+-----+ | 1 | 2019 | 113 | | 2 | 2008 | 121 | | 3 | 2009 | 12 | | 7 | 2018 | 0 | | 7 | 2019 | 0 | | 7 | 2020 | 30 | | 13 | 2019 | 40 | +-----+-----+

Explanation:

The npv value of (7, 2018) is not present in the NPV table, we consider it 0. The npv values of all other queries can be found in the NPV table.

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 npv_queries(npv: pd.DataFrame, queries: 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 npv_queries(npv: pd.DataFrame, queries: pd.DataFrame) -> pd.DataFrame:
```