1623. All Valid Triplets That Can Represent a Country

Description

Table: `SchoolA`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| student_id    | int     |
| student_name  | varchar |
+---------------+---------+
student_id is the column with unique values for this table.
Each row of this table contains the name and the id of a student in school A.
All student_name are distinct.
```

Table: `SchoolB`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| student_id    | int     |
| student_name  | varchar |
+---------------+---------+
student_id is the column with unique values for this table.
Each row of this table contains the name and the id of a student in school B.
All student_name are distinct.
```

Table: `SchoolC`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| student_id    | int     |
| student_name  | varchar |
+---------------+---------+
student_id is the column with unique values for this table.
Each row of this table contains the name and the id of a student in school C.
All student_name are distinct.
```

There is a country with three schools, where each student is enrolled in **exactly one** school. The country is joining a competition and wants to select one student from each school to represent the country such that:

- `member_A` is selected from `SchoolA`,
- `member_B` is selected from `SchoolB`,
- `member_C` is selected from `SchoolC`, and
- The selected students' names and IDs are pairwise distinct (i.e. no two students share the same name, and no two students share the same ID).

Write a solution to find all the possible triplets representing the country under the given constraints.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

```
Input:
SchoolA table:
+------------+--------------+
| student_id | student_name |
+------------+--------------+
| 1          | Alice        |
| 2          | Bob          |
+------------+--------------+
SchoolB table:
+------------+--------------+
| student_id | student_name |
+------------+--------------+
| 3          | Tom          |
+------------+--------------+
SchoolC table:
+------------+--------------+
| student_id | student_name |
+------------+--------------+
| 3          | Tom          |
| 2          | Jerry        |
| 10         | Alice        |
+------------+--------------+
Output:
+----------+----------+----------+
| member_A | member_B | member_C |
+----------+----------+----------+
| Alice    | Tom      | Jerry    |
| Bob      | Tom      | Alice    |
+----------+----------+----------+
Explanation:
Let us see all the possible triplets.
- (Alice, Tom, Tom) --> Rejected because member_B and member_C have the same name and the same ID.
- (Alice, Tom, Jerry) --> Valid triplet.
- (Alice, Tom, Alice) --> Rejected because member_A and member_C have the same name.
- (Bob, Tom, Tom) --> Rejected because member_B and member_C have the same name and the same ID.
- (Bob, Tom, Jerry) --> Rejected because member_A and member_C have the same ID.
- (Bob, Tom, Alice) --> Valid triplet.
```