SQL Schema

Table: SchoolA

+---------------+---------+

| Column Name | Type |

+---------------+---------+

| student\_id | int |

| student\_name | varchar |

+---------------+---------+

student\_id is the primary key 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 primary key 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 primary key 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 an SQL query to find all the possible triplets representing the country under the given constraints.

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

The query result format is in the following example.

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 |

+------------+--------------+

Result table:

+----------+----------+----------+

| member\_A | member\_B | member\_C |

+----------+----------+----------+

| Alice | Tom | Jerry |

| Bob | Tom | Alice |

+----------+----------+----------+

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.