SQL Schema

Table: Friendship

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

| Column Name | Type |

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

| user1\_id | int |

| user2\_id | int |

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

(user1\_id, user2\_id) is the primary key for this table.

Each row of this table indicates that the users user1\_id and user2\_id are friends.

Note that user1\_id < user2\_id.

A friendship between a pair of friends x and y is **strong** if x and y have **at least three** common friends.

Write an SQL query to find all the **strong friendships**.

Note that the result table should not contain duplicates with user1\_id < user2\_id.

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

The query result format is in the following example:

Friendship table:

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

| user1\_id | user2\_id |

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

| 1 | 2 |

| 1 | 3 |

| 2 | 3 |

| 1 | 4 |

| 2 | 4 |

| 1 | 5 |

| 2 | 5 |

| 1 | 7 |

| 3 | 7 |

| 1 | 6 |

| 3 | 6 |

| 2 | 6 |

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

Result table:

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

| user1\_id | user2\_id | common\_friend |

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

| 1 | 2 | 4 |

| 1 | 3 | 3 |

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

Users 1 and 2 have 4 common friends (3, 4, 5, and 6).

Users 1 and 3 have 3 common friends (2, 6, and 7).

We did not include the friendship of users 2 and 3 because they only have two common friends (1 and 6).