Table: Contests

```
+--------------+------+
| Column Name  | Type |
+--------------+------+
| contest_id   | int  |
| gold_medal   | int  |
| silver_medal | int  |
| bronze_medal | int  |
+--------------+------+
contest_id is the primary key for this table.
This table contains the LeetCode contest ID and the user IDs of the gold, silver, and bronze
It is guaranteed that any consecutive contests have consecutive IDs and that no ID is skippe
```

Table: ``Users``

Column Name	Type
user_id mail	int
name	varchar
	varchar

user_id is the primary key for this table. This table contains information about the users.

```
Write an SQL query to report the ``name`` and the ``mail`` of all **interview candidates**.

<ul>
    <li>The user won **any** medal in **three or more consecutive** contests.</li>
    <li>The user won the **gold** medal in **three or more different** contests (not necessa
</ul>

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

The query result format is in the following example.


<strong class="example">Example 1:**
```

Input: Contests table: +————+————+————+————+ | contest_id | gold_medal | silver_medal | bronze_medal | +————+———— +————+————+ | 190 | 1 | 5 | 2 | | 191 | 2 | 3 | 5 | | 192 | 5 | 2 | 3 | | 193 | 1 | 3 | 5 | | 194 | 4 | 5 | 2 | | 195 | 4 | 2 | 1 | | 196 | 1 | 5 | 2 | +————+————+— ————+————+ Users table: +———+—————+——-+ | user_id | mail | name | +———+—————+——-+ | 1 | sarah@leetcode.com | Sarah | | 2 | bob@leetcode.com | Bob | | 3 | alice@leetcode.com | Alice | | 4 | hercy@leetcode.com | Hercy | | 5 | quarz@leetcode.com | Quarz | +———+— ————+——-+ Output: +——-+————+ | name | mail | +——-+————+ | Sarah | sarah@leetcode.com | | Bob | bob@leetcode.com | | Alice | alice@leetcode.com | | Quarz | quarz@leetcode.com | +——-+——— ———+ Explanation: Sarah won 3 gold medals (190, 193, and 196), so we include her in the result table. Bob won a medal in 3 consecutive contests (190, 191, and 192), so we include him in the result table. - Note that he also won a medal in 3 other consecutive contests (194, 195, and 196). Alice won a medal in 3 consecutive contests (191, 192, and 193), so we include her in the result table. Quarz won a medal in 5 consecutive contests (190, 191, 192, 193, and 194), so we include them in the result table. “‘

Follow up:

What if the first condition changed to be “any medal in `n`** or more** consecutive contests”? How would you change your solution to get the interview candidates? Imagine that `n` is the parameter of a stored procedure.

Some users may not participate in every contest but still perform well in the ones they do. How would you change your solution to only consider contests where the user **was a participant**? Suppose the registered users for each contest are given in another table.