Run Submit

1939. Users That Actively Request Confirmation Messages `Premium`

Solved ⊘

`Easy` 🏷 Topics

SQL Schema > Pandas Schema >

Table: `Signups`

```
+-------------+----------+
| Column Name | Type     |
+-------------+----------+
| user_id     | int      |
| time_stamp  | datetime |
+-------------+----------+
```

user_id is the column with unique values for this table.
Each row contains information about the signup time for the user with ID user_id.

Table: `Confirmations`

```
+-------------+----------+
| Column Name | Type     |
+-------------+----------+
| user_id     | int      |
| time_stamp  | datetime |
| action      | ENUM     |
+-------------+----------+
```

(user_id, time_stamp) is the primary key (combination of columns with unique values) for this table.
user_id is a foreign key (reference column) to the Signups table.
action is an ENUM (category) of the type ('confirmed', 'timeout')
Each row of this table indicates that the user with ID user_id requested a confirmation message at time_stamp and that confirmation message was either confirmed ('confirmed') or expired without confirming ('timeout').

Write a solution to find the IDs of the users that requested a confirmation message **twice** within a 24-hour window. Two messages exactly 24 hours apart are considered to be within the window. The `action` does not affect the answer, only the request time.

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

The result format is in the following example.

Example 1:

Input:
Signups table:
```
+---------+---------------------+
| user_id | time_stamp          |
+---------+---------------------+
| 3       | 2020-03-21 10:16:13 |
| 7       | 2020-01-04 13:57:59 |
| 2       | 2020-07-29 23:09:44 |
| 6       | 2020-12-09 10:39:37 |
+---------+---------------------+
```
Confirmations table:
```
+---------+---------------------+-----------+
| user_id | time_stamp          | action    |
+---------+---------------------+-----------+
| 3       | 2021-01-06 03:30:46 | timeout   |
| 3       | 2021-01-06 03:37:45 | timeout   |
| 7       | 2021-06-12 11:57:29 | confirmed |
| 7       | 2021-06-13 11:57:30 | confirmed |
| 2       | 2021-01-22 00:00:00 | confirmed |
| 2       | 2021-01-23 00:00:00 | timeout   |
| 6       | 2021-10-23 14:14:14 | confirmed |
| 6       | 2021-10-24 14:14:13 | timeout   |
+---------+---------------------+-----------+
```

Output:
```
+---------+
| user_id |
+---------+
| 2       |
| 3       |
| 6       |
+---------+
```

Explanation:
User 2 requested two messages within exactly 24 hours of each other, so we include them.
User 3 requested two messages within 6 minutes and 59 seconds of each other, so we include them.
User 6 requested two messages within 23 hours, 59 minutes, and 59 seconds of each other, so we include them.
User 7 requested two messages within 24 hours and 1 second of each other, so we exclude them from the answer.

Seen this question in a real interview before?   1/5
Yes   No

Accepted **10.9K** | Submissions **19.3K** | Acceptance Rate **56.3%**

🏷 Topics   ⌄

💬 Discussion (3)   ⌄

👍 61 👎 💬 3 ⭐ ↗ ?

Code

MySQL ⌄   • Auto

```sql
1  # Write your MySQL query statement below
2
3
4  SELECT DISTINCT a.user_id FROM Confirmations a, Confirmations b
5  WHERE a.user_id = b.user_id AND b.time_stamp - a.time_stamp <= 1000000 AND b.time_stamp > a.time_stamp
```

☁ Saved   Ln 5, Col 104

☑ Testcase >_ Test Result

Accepted Runtime: 370 ms

• Case 1

Input

Signups =
```
| user_id | time_stamp          |
| ------- | ------------------- |
| 3       | 2020-03-21 10:16:13 |
| 7       | 2020-01-04 13:57:59 |
| 2       | 2020-07-29 23:09:44 |
| 6       | 2020-12-09 10:39:37 |
```

Confirmations =
```
| user_id | time_stamp          | action    |
| ------- | ------------------- | --------- |
| 3       | 2021-01-06 03:30:46 | timeout   |
```

```
| 3       | 2021-01-06 03:37:45 | timeout   |
```