2990. Loan Types

Description

Table: `Loans`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| loan_id     | int     |
| user_id     | int     |
| loan_type   | varchar |
+-------------+---------+
loan_id is column of unique values for this table.
This table contains loan_id, user_id, and loan_type.
```

Write a solution to find all **distinct** `user_id`'s that have **at least one** **Refinance** loan type and at least one **Mortgage** loan type.

Return *the result table ordered by* `user_id` *in* **ascending** *order* .

The result format is in the following example.

Example 1:

```
Input:
Sessions table:
+---------+---------+-----------+
| loan_id | user_id | loan_type |
+---------+---------+-----------+
| 683     | 101     | Mortgage  |
| 218     | 101     | AutoLoan  |
| 802     | 101     | Inschool  |
| 593     | 102     | Mortgage  |
| 138     | 102     | Refinance |
| 294     | 102     | Inschool  |
| 308     | 103     | Refinance |
| 389     | 104     | Mortgage  |
+---------+---------+-----------+
Output
+---------+
| user_id |
+---------+
| 102     |
+---------+
Explanation
- User_id 101 has three loan types, one of which is a Mortgage. However, this user does not have any loan type categorized as Refinance, so user_id
101 won't be considered.
- User_id 102 possesses three loan types: one for Mortgage and one for Refinance. Hence, user_id 102 will be included in the result.
- User_id 103 has a loan type of Refinance but lacks a Mortgage loan type, so user_id 103 won't be considered.
- User_id 104 has a Mortgage loan type but doesn't have a Refinance loan type, thus, user_id 104 won't be considered.
Output table is ordered by user_id in ascending order.
```