Table: Movies

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

| Column Name | Type |

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

| movie\_id | int |

| title | varchar |

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

movie\_id is the primary key for this table.

title is the name of the movie.

Table: Users

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

| Column Name | Type |

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

| user\_id | int |

| name | varchar |

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

user\_id is the primary key for this table.

Table: Movie\_Rating

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

| Column Name | Type |

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

| movie\_id | int |

| user\_id | int |

| rating | int |

| created\_at | date |

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

(movie\_id, user\_id) is the primary key for this table.

This table contains the rating of a movie by a user in their review.

created\_at is the user's review date.

Write the following SQL query:

* Find the name of the user who has rated the greatest number of movies.

In case of a tie, return lexicographically smaller user name.

* Find the movie name with the ***highest average*** rating in **February 2020**.

In case of a tie, return lexicographically smaller movie name.

The query is returned in 2 rows, the query result format is in the following example:

Movies table:

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

| movie\_id | title |

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

| 1 | Avengers |

| 2 | Frozen 2 |

| 3 | Joker |

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

Users table:

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

| user\_id | name |

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

| 1 | Daniel |

| 2 | Monica |

| 3 | Maria |

| 4 | James |

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

Movie\_Rating table:

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

| movie\_id | user\_id | rating | created\_at |

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

| 1 | 1 | 3 | 2020-01-12 |

| 1 | 2 | 4 | 2020-02-11 |

| 1 | 3 | 2 | 2020-02-12 |

| 1 | 4 | 1 | 2020-01-01 |

| 2 | 1 | 5 | 2020-02-17 |

| 2 | 2 | 2 | 2020-02-01 |

| 2 | 3 | 2 | 2020-03-01 |

| 3 | 1 | 3 | 2020-02-22 |

| 3 | 2 | 4 | 2020-02-25 |

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

Result table:

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

| results |

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

| Daniel |

| Frozen 2 |

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

Daniel and Monica have rated 3 movies ("Avengers", "Frozen 2" and "Joker") but Daniel is smaller lexicographically.

Frozen 2 and Joker have a rating average of 3.5 in February but Frozen 2 is smaller lexicographically.