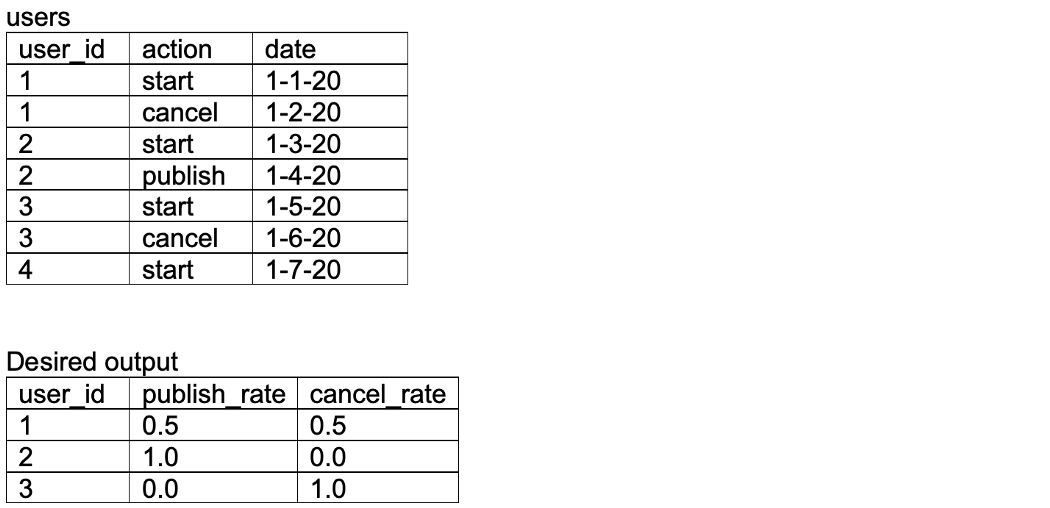
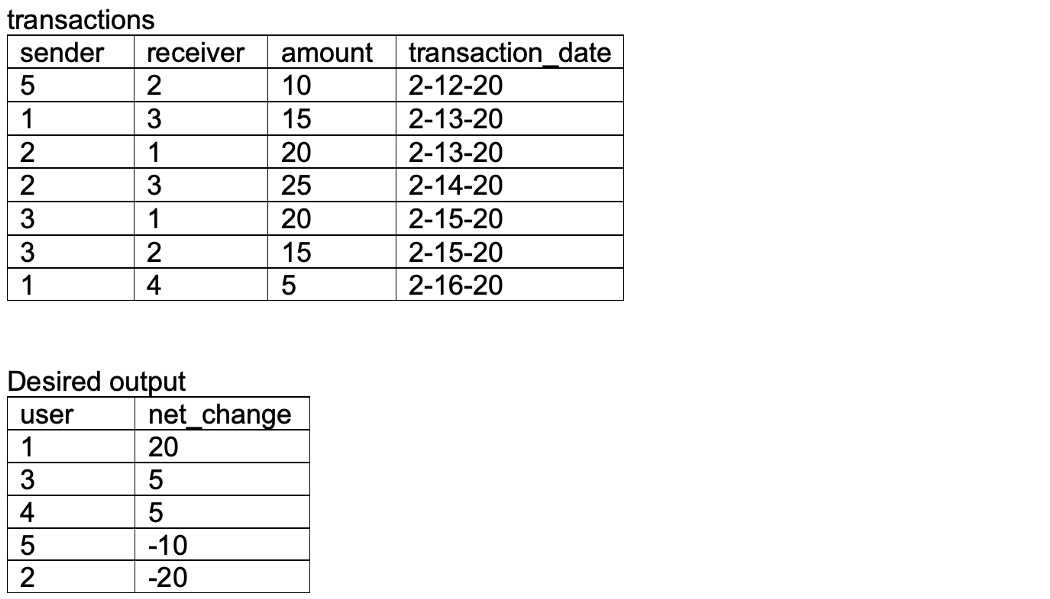
1. Cancellation rates

From the following table of user IDs, actions, and dates, write a query to return the publication and cancellation rate for each user.



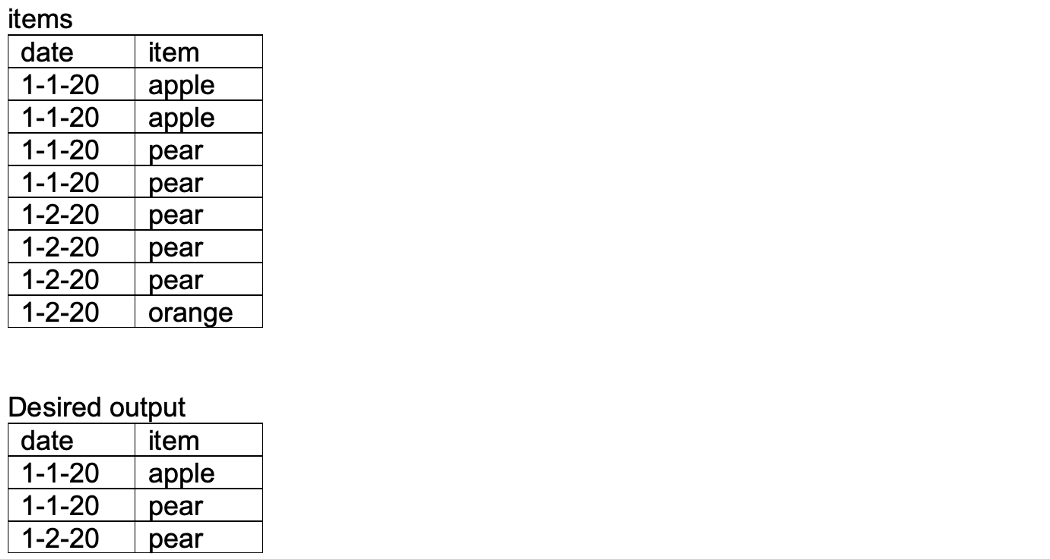
2. Changes in net worth

From the following table of transactions between two users, write a query to return the change in net worth for each user, ordered by decreasing net change.



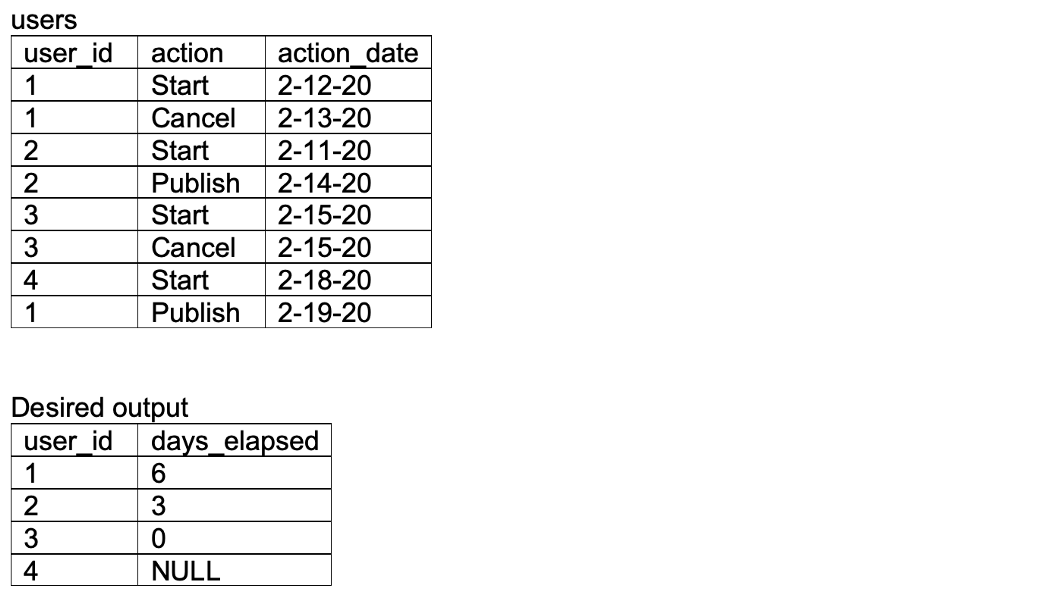
## 3. Most frequent items

From the following table containing a list of dates and items ordered, write a query to return the most frequent item ordered on each date. Return multiple items in the case of a tie.



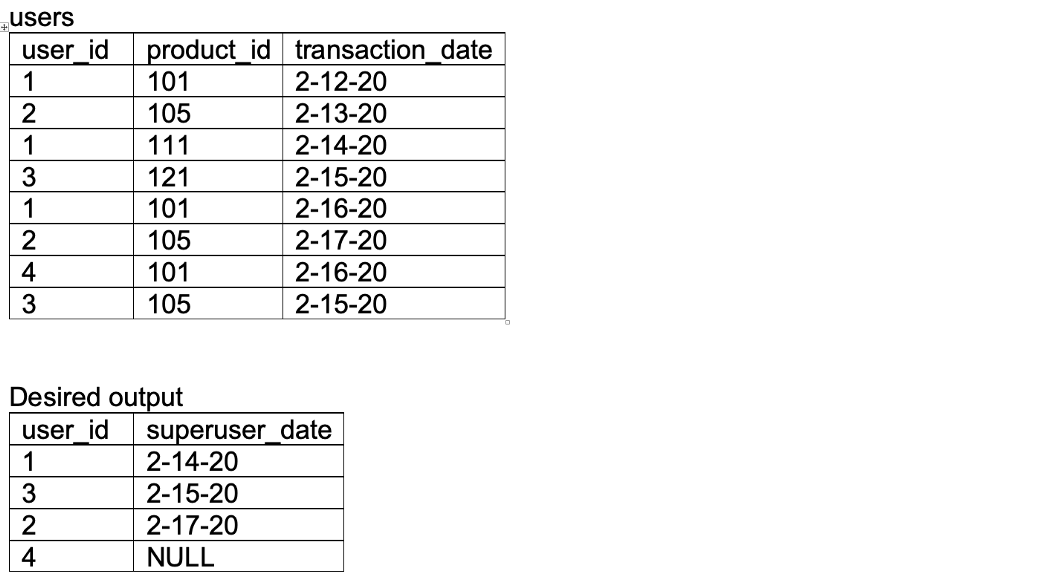
## 4. Time difference between latest actions

From the following table of user actions, write a query to return for each user the time elapsed between the last action and the second-to-last action, in ascending order by user ID.



5. Super users

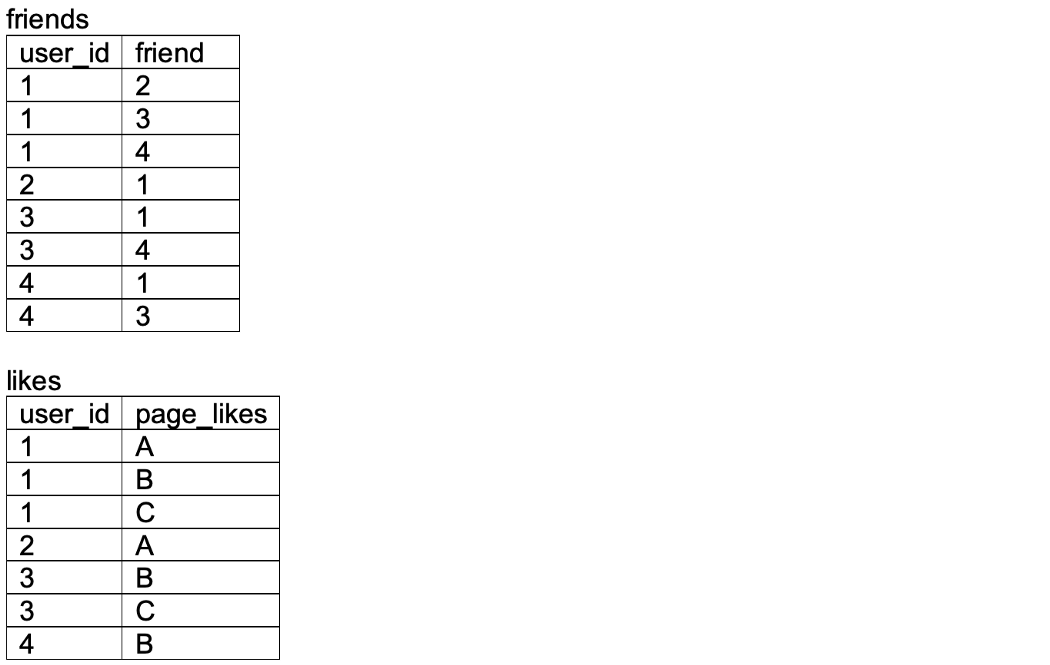
A company defines its super users as those who have made at least two transactions. From the following table, write a query to return, for each user, the date when they become a super user, ordered by oldest super users first. Users who are not super users should also be present in the table.



## 

## 6. Content recommendation (hard)

Using the following two tables, write a query to return page recommendations to a social media user based on the pages that their friends have liked, but that they have not yet marked as liked. Order the result by ascending user ID. [Source](https://www.glassdoor.com/Interview/Write-an-SQL-query-that-makes-recommendations-using-the-pages-that-your-friends-liked-Assume-you-have-two-tables-a-two-c-QTN_1413464.htm).

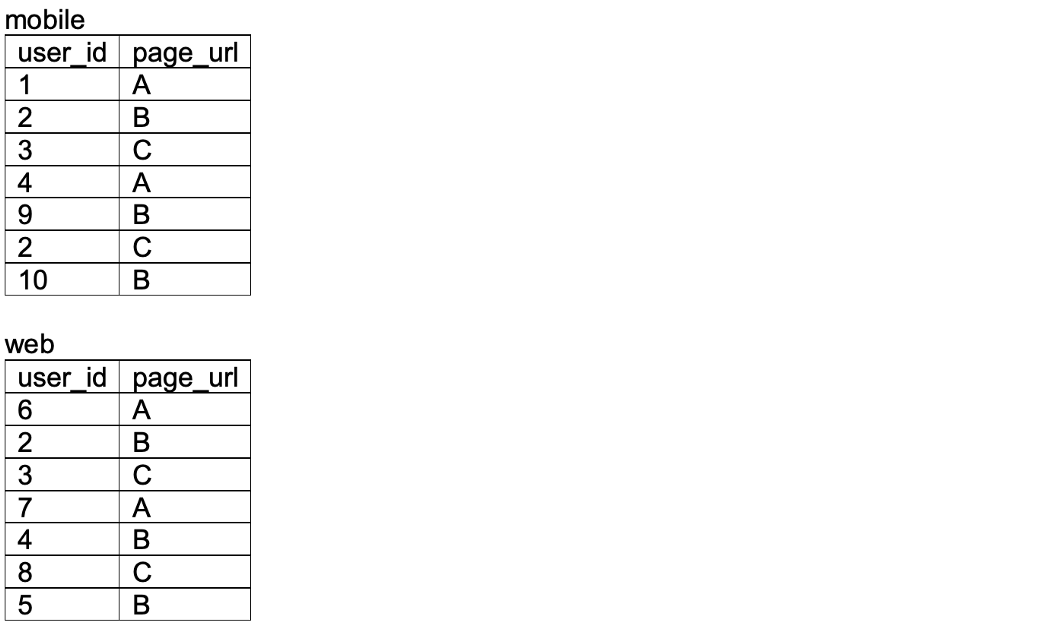


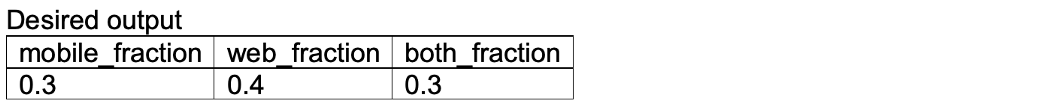


## 

## 7. Mobile and web visitors

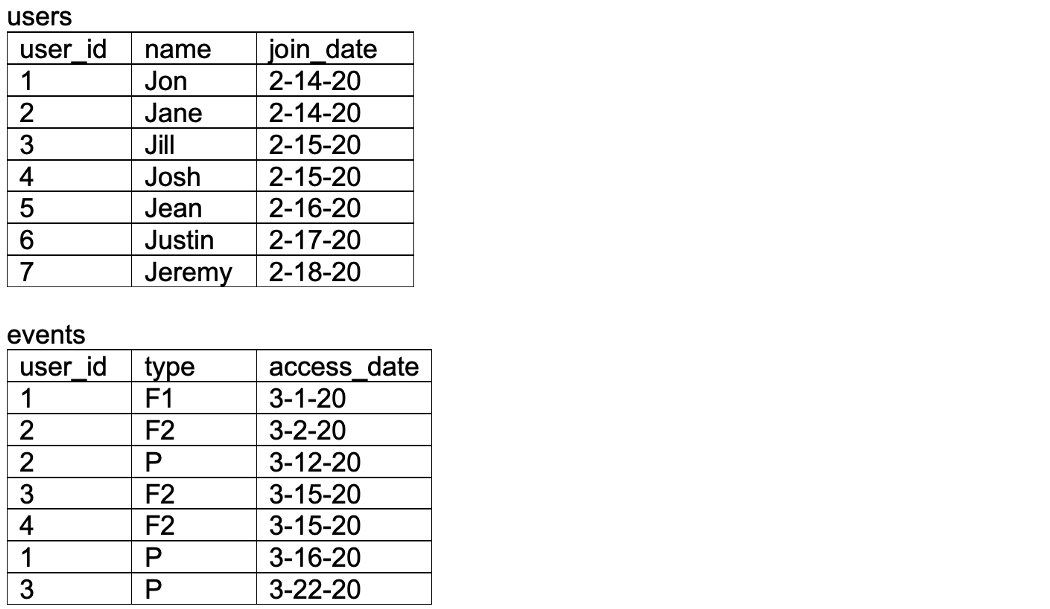
With the following two tables, return the fraction of users who only visited mobile, only visited web, and visited both.





8. Upgrade rate by product action (hard)

Given the following two tables, return the fraction of users, rounded to two decimal places, who accessed feature two (type: F2 in events table) and upgraded to premium within the first 30 days of signing up.

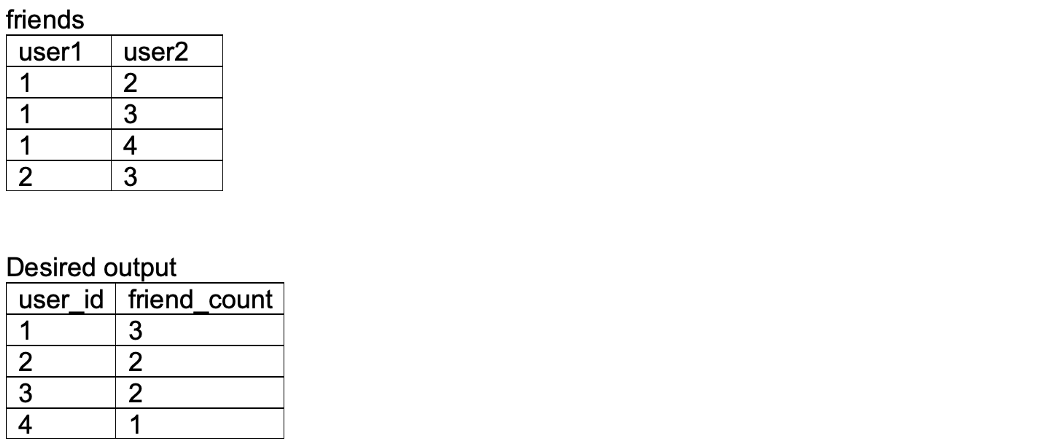




## 

## 9. Most friended

Given the following table, return a list of users and their corresponding friend count. Order the result by descending friend count, and in the case of a tie, by ascending user ID. Assume that only unique friendships are displayed



## 

## 10. Project aggregation

The projects table contains three columns: task\_id, start\_date, and end\_date. The difference between end\_date and start\_date is 1 day for each row in the table. If task end dates are consecutive they are part of the same project. Projects do not overlap.

Write a query to return the start and end dates of each project, and the number of days it took to complete. Order by ascending project duration, and descending start date in the case of a tie. From

