6 9 6

597. Friend Requests I: Overall Acceptance Rate

□ Articles > 597. Friend Requests I: Overall Acceptance Rate ▼

June 9, 2017 | 31.7K views

Average Rating: 4.50 (20 votes)

In social network like Facebook or Twitter, people send friend requests and accept others' requests as well. Now given two tables as below:

Table: friend_request

	1	1	2	2016_06-01
	1	1	3	2016_06-01
1	1	1	4	2016_06-01
	2	1	3	2016_06-02
Ĩ	3	-	4	2016-06-09

| requester_id | accepter_id |accept_date |

acceptance divide the number of requests.

Table: request_accepted

```
|-----|
               2016_06-03
              2016-06-08
              2016-06-08
3
               2016-06-09
               2016-06-10
```

For the sample data above, your query should return the following result.

Write a query to find the overall acceptance rate of requests rounded to 2 decimals, which is the number of

accept_rate

```
|----|
 0.80
```

Note:

divide it by the number of requests to get the acceptance rate. • It is possible that a sender sends multiple requests to the same receiver, and a request could be accepted more than once. In this case, the 'duplicated' requests or acceptances are only counted once. If there is no requests at all, you should return 0.00 as the accept_rate.

 The accepted requests are not necessarily from the table friend_request. In this case, you just need to simply count the total accepted requests (no matter whether they are in the original requests), and

- **Explanation:** There are 4 unique accepted requests, and there are 5 requests in total. So the rate is 0.80.
- Follow-up:

 Can you write a query to return the accept rate but for every month? How about the cumulative accept rate for every day?

Solution

```
Intuition
```

Approach: Using round and ifnull [Accepted]

Algorithm

rate.

To get the distinct number of accepted requests, we can query from the request_accepted table.

Count the accepted requests and then divides it by the number of all requests.

```
With the same technique, we can have the total number of requests from the friend_request table:
  select count(*) from (select distinct sender_id, send_to_id from friend_request;
```

At last, divide these two numbers and round it to a scale of 2 decimal places to get the required acceptance

select count(*) from (select distinct requester_id, accepter_id from request_accepted;

utilize ifnull to deal with this special case. MySQL

Wait! The divisor (total number of requests) could be '0' if the table **friend_request** is empty. So, we have to

select round(

(select count(*) from (select distinct requester_id, accepter_id from request_acce

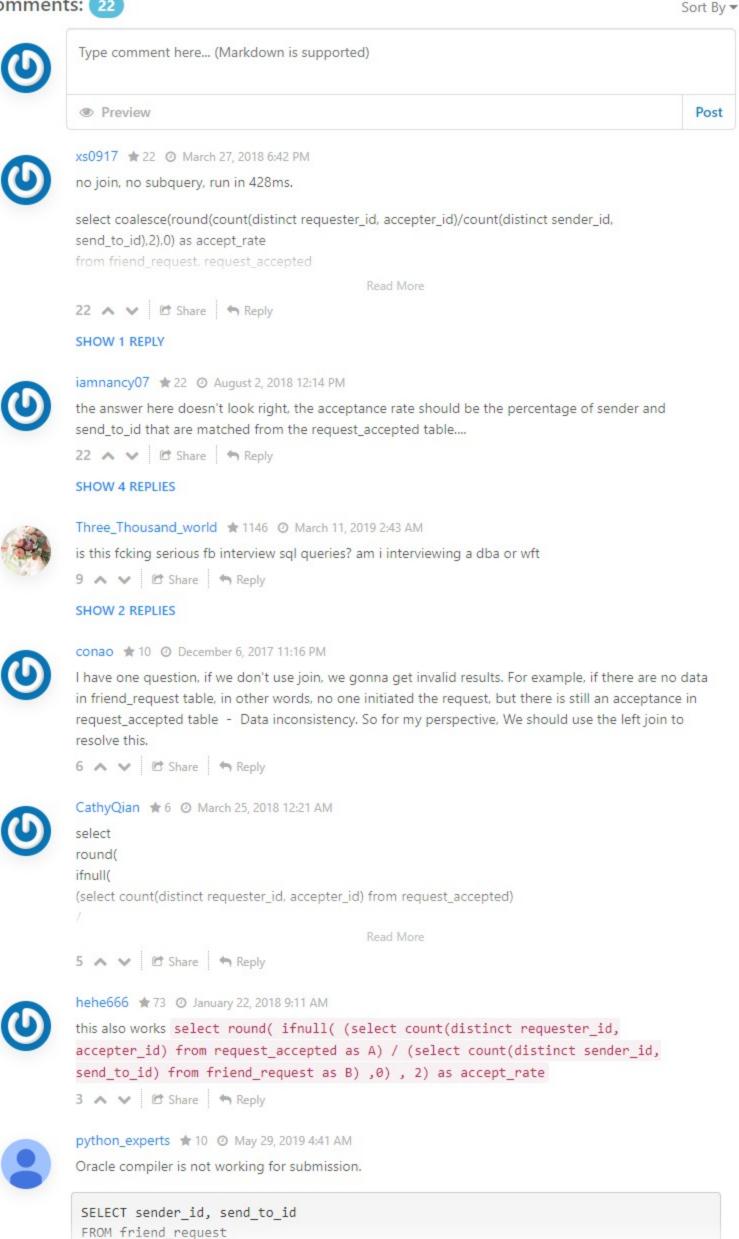
Next **⊙**

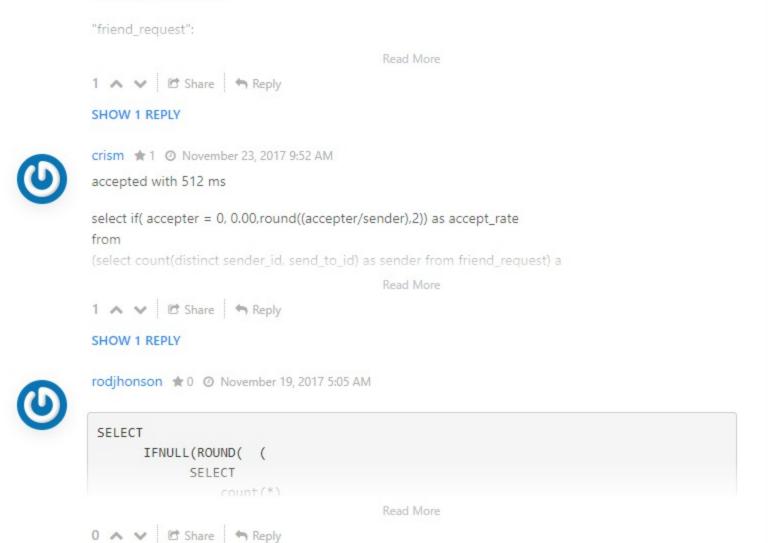
```
(select count(*) from (select distinct sender_id, send_to_id from friend_request)
, 2) as accept_rate;
```

Comments: 22

Rate this article: * * * * *

O Previous





Read More

oroun hy sender id send to id

is sender_id = requester_id and send_to_id = accepter_id?

2 A V C Share Reply

Here is one test cases:

(123)