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`

```
| sender_id | send_to_id |request_date|
|-----------|------------|------------|
| 1         | 2          | 2016_06-01 |
| 1         | 3          | 2016_06-01 |
| 1         | 4          | 2016_06-01 |
| 2         | 3          | 2016_06-02 |
| 3         | 4          | 2016-06-09 |
```

Table: `request_accepted`

```
| requester_id | accepter_id |accept_date |
|--------------|-------------|------------|
| 1            | 2           | 2016_06-03 |
| 1            | 3           | 2016-06-08 |
| 2            | 3           | 2016-06-08 |
| 3            | 4           | 2016-06-09 |
| 3            | 4           | 2016-06-10 |
```

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

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

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

Note:

- 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 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.

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

Approach: Using `round` and `ifnull` [Accepted]

Intuition

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

Algorithm

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

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

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 rate.

Wait! The divisor (total number of requests) could be '0' if the table **friend_request** is empty. So, we have to utilize `ifnull` to deal with this special case.

MySQL

```
select
round(
    ifnull(
    (select count(*) from (select distinct requester_id, accepter_id from request_acce
    /
    (select count(*) from (select distinct sender_id, send_to_id from friend_request)
    0)
, 2) as accept_rate;
```

Rate this article:

Previous Next

Comments: 22

Sort By

Type comment here... (Markdown is supported)

Preview Post

xs0917 ★ 22 March 27, 2018 6:42 PM

no join, no subquery, run in 428ms.

```
select coalesce(round(count(distinct requester_id, accepter_id)/count(distinct sender_id,
send_to_id),2),0) as accept_rate
from friend_request, request_accepted
```

Read More

22 Share Reply

SHOW 1 REPLY

iamnancy07 ★ 22 August 2, 2018 12:14 PM

the answer here doesn't look right. the acceptance rate should be the percentage of sender and send_to_id that are matched from the request_accepted table....

22 Share Reply

SHOW 4 REPLIES

Three_Thousand_world ★ 1146 March 11, 2019 2:43 AM

is this fcking serious fb interview sql queries? am i interviewing a dba or wft

9 Share Reply

SHOW 2 REPLIES

conao ★ 10 December 6, 2017 11:16 PM

I have one question. if we don't use join, we gonna get invalid results. For example, if there are no data in friend_request table. in other words, no one initiated the request. but there is still an acceptance in request_accepted table - Data inconsistency. So for my perspective. We should use the left join to resolve this.

6 Share Reply

CathyQian ★ 6 March 25, 2018 12:21 AM

select
round(
ifnull(
(select count(distinct requester_id, accepter_id) from request_accepted)

Read More

5 Share Reply

hehe666 ★ 73 January 22, 2018 9:11 AM

this also works `select round( ifnull( (select count(distinct requester_id, accepter_id) from request_accepted as A) / (select count(distinct sender_id, send_to_id) from friend_request as B) ,0) , 2) as accept_rate`

3 Share Reply

python_experts ★ 10 May 29, 2019 4:41 AM

Oracle compiler is not working for submission.

```
SELECT sender_id, send_to_id
FROM friend_request
```

Read More

2 Share Reply

annayu12 ★ 13 January 9, 2019 3:29 AM

is sender_id = requester_id and send_to_id = accepter_id?

Here is one test cases:

"friend_request":

Read More

1 Share Reply

SHOW 1 REPLY

crism ★ 1 November 23, 2017 9:52 AM

accepted with 512 ms

```
select if( accepter = 0, 0.00,round((accepter/sender),2)) as accept_rate
from
(select count(distinct sender_id, send_to_id) as sender from friend_request) a
```

Read More

1 Share Reply

SHOW 1 REPLY

rodjhonson ★ 0 November 19, 2017 5:05 AM

```
SELECT
    IFNULL(ROUND(  (
        SELECT
```

Read More

0 Share Reply