Instructions

Read through this document, studying the tables and ERD. Then, write SQL queries that answer the questions listed in the SQL Queries section. Finally, submit your answers.

The following questions contain a series of tables taken from the social media site FindingFastFriends.

TablesOverview

You can see all the tables and their descriptions listed below. Each of these tables contain 1M+ rows of data. You can see the first three rows of each table to get a sense of what they contain. The tables necessary to answer each individual question will be repeated throughout the assessment for you to reference.

users			
user_id	username	email	friend_count
3437315	janedoe2.0	janedoe2.0@email.com	5
3437316	princessluv8996	princessluv8996@email.com	600017
3437317	sk8ter4l!f3	sk8ter4l!f3@email.com	583

TableName:*users*
user_id: *Id of user*
username: *name of user*
email: *email of user*
friend_count: *user's number of friends*

friend_requests				
action_id	requester_id	requestee_id	action_timestamp	action_taken
1	1037392	3437315	2015-03-1500:01:05	Requested
2	2138102	5438443	2015-03-1500:01:07	Accepted
3	2331234	1231232	2015-03-1500:01:08	Rejected

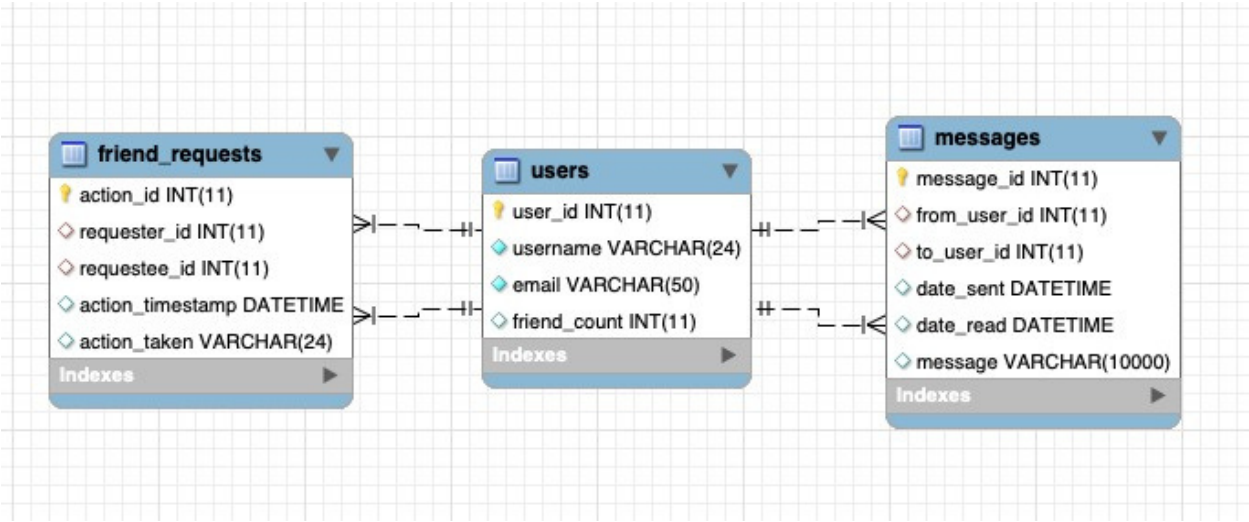
Table Name: *friend_requests*
action_id: *Id of action*
requester_id: *Id of user who took action*
requestee_id: *Id of user who had action taken on*
action_timestamp: *Timestamp of user A action*
action_taken: *Type of action user took (Requested, Accepted, Rejected)*

messages					
message_id	**from_user_id**	**to_user_id**	**date_sent**	**date_read**	**message**
1	3437317	2138102	2015-03-15 00:02:17	2015-0315 00:03:05	"Hi!!! Wanted to know how u were doing?! Miss you !! ..."
2	1438443	5937440	2015-03-15 00:02:24	NULL	"Don't forget to like and comment on my new pic..."
3	2331234	1231232	2015-03-15 00:02:25	2015-04-01 00:11:08	"Let's hang out!"

TableName:*messages*
message_id: *Id of message*
from_user_id: *Id of user sending the message*
to_user_id: *Id of user receiving the message*
date_sent: *date that the message was sent*
date_read: *date that the message was read. NULL if message has not been read.*
message: *text of the message*

ERD

Here is the ERD of the database for your reference.



SQL Queries

You have `users` table containing 1M+ rows of user information. Below are the first 3 a rows.

users			
user_id	username	email	friend_count
3437315	janedoe2.0	janedoe2.0@email.com	5
3437316	princessluv8996	princessluv8996@email.com	600017
3437317	sk8ter4l!f3	sk8ter4l!f3@email.com	583

Question 1: Write a SQL query that returns the email address and friend count of the user with the most friends.

HINT: What if more than one user has the same "max" friend count?

```
SELECT email, friend_count
FROM users
WHERE friend_count = (
SELECT MAX(friend_count) FROM users
);
```

You have a users table containing 1M+ rows of user information and a friend_requests table containing 1M+ rows of friend request information. Below are the first 3 rows of each table.

users			
user_id	username	email	friend_count
3437315	janedoe2.0	janedoe2.0@email.com	5
3437316	princessluv8996	princessluv8996@email.com	600017
3437317	sk8ter4l!f3	sk8ter4l!f3@email.com	583

friend_requests				
action_id	requester_id	**requestee_id**	**action_timestamp**	action_taken
1	1037392	3437315	2015-03-15 00:01:05	Requested
2	2138102	5438443	2015-03-15 00:01:07	Accepted
3	2331234	1231232	2015-03-15 00:01:08	Rejected

Question 2: Write a SQL query that returns the three users who have sent the most friend requests. Your query should return the username and number of requests sent.

```
SELECT users.username, COUNT(fr.requestee_id) AS num_requests_sent
FROM users
JOIN friend_requests AS fr ON users.user_id = fr.requester_id
GROUP BY fr.requester_id, users.username
ORDER BY num_requests_sent DESC
LIMIT 3;
```

Consider the friend_requests table again. It contains a column `action_taken` that shows whether a friend request was: Requested, Accepted or Rejected.

friend_requests				
action_id	requester_id	**requestee_id**	**action_timestamp**	action_taken
1	1037392	3437315	2015-03-15 00:01:05	Requested
2	2138102	5438443	2015-03-15 00:01:07	Accepted
3	2331234	1231232	2015-03-15 00:01:08	Rejected

Question 3A: Write a query to determine the number of Accepted friend requests.

```
SELECT COUNT(*)
FROM friend_requests
WHERE action_taken = 'Accepted';
```

Question 3B: Write a second query to determine the percentage of requests that are Accepted.

```
SELECT (COUNT(*) * 100 / (SELECT COUNT(*) FROM friend_requests)) AS percentage_accepted
FROM friend_requests
WHERE action_taken = 'Accepted';
```

The database also has a third table messages, that includes 1M+ rows of information related to the messages sent between users. Here are the first three rows.

messages					
message_id	**from_user_id**	**to_user_id**	**date_sent**	**date_read**	**message**
1	3437317	2138102	2015-03-15 00:02:17	2015-0315 00:03:05	"Hi!!! Wanted to know how u were doing?! Miss you !! ..."
2	1438443	5937440	2015-03-15 00:02:24	NULL	"Don't forget to like and comment on my new pic..."
3	2331234	1231232	2015-03-15 00:02:25	2015-04-01 00:11:08	"Let's hang out!"

Question 4: Write a SQL query to count the number of messages that include the following phrase: "Miss you" (Note: You should account for a capital "M" and lowercase "m".)

```
SELECT COUNT(*) AS num_messages
FROM messages
WHERE LOWER(message) LIKE '%miss
you%';
```

Consider the following `users` and `messages` tables to answer the question below.

users			
user_id	username	email	friend_count
3437315	janedoe2.0	janedoe2.0@email.com	5
3437316	princessluv8996	princessluv8996@email.com	600017
3437317	sk8ter4l!f3	sk8ter4l!f3@email.com	583

messages					
message_id	from_user_id	to_user_id	date_sent	date_read	message
1	3437317	2138102	2015-03-15 00:02:17	2015-0315 00:03:05	"Hi!!! Wanted to know how u were doing?! Miss you !! ..."
2	1438443	5937440	2015-03-15 00:02:24	NULL	"Don't forget to like and comment on my new pic..."
3	2331234	1231232	2015-03-15 00:02:25	2015-04-01 00:11:08	"Let's hang out!"

Question 5: Write a SQL query to determine which users have more than 10 unread messages.

```
SELECT from_user_id, COUNT(*) AS
unread_messages FROM messages
WHERE date_read IS NULL
GROUP BY from_user_id
HAVING COUNT(*) > 10;
```