

Now, there are two rules for this feature

1. If user follows the searching poster, and be accepted. Then user only can see the pictures that are allowed followers to see.
2. If user and poster are in the same group, and user can see all the visible photo in the ShareWith table in that group and posted by the search poster.

First, user are able to see the search function in main page

Homepage

Welcome Back, Zhenggg!

Here are your photos:

Photo	Time Posted
6	2019-12-07 19:44:39
7	2019-12-07 19:44:51

[Upload a Photo](#)
[View All Visible Photos!](#)
[View a Profile](#)
[Follow Requests](#)
[Find Poster Picture](#)

[Logout](#)

If user go into find poster picture, then it is able to see

Who you want to find

Once user types in name and submits, system will do the query.

Now, we have two users, eleven and eleven2.

For first part

Eleven2 is following the eleven, and eleven accepts that by follow status.

```
mysql> select * from Comments;
+-----+-----+-----+-----+
| username | photoID | comments | commenttime |
+-----+-----+-----+-----+
| eleven | 5 | test | 2019-12-05 15:42:48 |
| eleven | 5 | again | 2019-12-05 15:42:54 |
| eleven | 5 | est | 2019-12-05 15:43:00 |
| eleven2 | 5 | wow, what a nice pict | 2019-12-11 15:00:37 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from Following;
ERROR 1146 (42S02): Table 'finstagram.following' doesn't exist
mysql> select * from Follow;
+-----+-----+-----+
| username_followed | username_follower | followstatus |
+-----+-----+-----+
| bobby | abby | 1 |
| bobby | colleen | 0 |
| eleven | eleven2 | 1 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

This means that eleven2 is able to see eleven's photo if that photo is open to all follower.

Let's see which photo of eleven is open to all follower

```

mysql> select * from photo
-> ;
+-----+-----+-----+-----+-----+
| photoID | postingdate | filepath | allFollowers | caption | photoPoster |
+-----+-----+-----+-----+-----+
| 1 | 2019-12-04 06:00:00 | /roommates_b.jpg | 1 | roommates | bobby |
| 2 | 2019-12-04 06:00:00 | /roommates_a.jpg | 1 | roommates | abby |
| 3 | 2019-12-04 06:00:00 | /bowling_team.jpg | 0 | bowlingTeam | bobby |
| 4 | 2019-12-04 06:00:00 | /family_bora_bora.jpg | 0 | family vaca | abby |
| 5 | 2019-12-04 10:12:52 | WechatIMG11.jpeg | 1 | NULL | eleven |
| 6 | 2019-12-07 19:44:39 | 35328.jpg | 1 | NULL | eleven2 |
| 7 | 2019-12-07 19:44:51 | 35328.jpg | 1 | NULL | eleven2 |
| 8 | 2019-12-08 11:38:40 | 8.jpeg | 1 | NULL | eleven |
| 9 | 2019-12-08 11:38:50 | Professional-Licensure-for-Computer-Engineers-and-Software-Engineers.jpg | 1 | NULL | eleven |
| 10 | 2019-12-08 11:39:01 | WechatIMG11.jpeg | 1 | NULL | eleven |
| 11 | 2019-12-08 11:39:14 | bafang.jpg | 0 | NULL | eleven |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>

```

As we can see, photo 5, 8, 9, and 10 are open to all follower, but 11 is not open to all follower posted by eleven.

Then if eleven2 search eleven,

This is pic post by your search poster

Photo	Time Posted
5	2019-12-05 10:12:52
8	2019-12-08 11:38:40
9	2019-12-08 11:38:50
10	2019-12-08 11:39:01

[Back](#)

Only 5, 8, 9, 10 are able to see! And 11 is not showing up. So part 1 is success.

For part 2

now, since eleven is not following eleven2, we can use this to test if eleven can search eleven2 in condition 2.

First, we made eleven and eleven2 as in the same group:

```

mysql> select * from BelongTo
-> ;
+-----+-----+-----+
| member_username | owner_username | groupName |
+-----+-----+-----+
| abby           | abby           | family      |
| dan            | abby           | family      |
| abby           | abby           | roommates   |
| coileen         | abby           | roommates   |
| bobby           | bobby           | roommates   |
| dan            | bobby           | roommates   |
| eleven          | eleven2        | test        |
| eleven2         | eleven2        | test        |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

Now, eleven and eleven2 in the same group as that owner is eleven2 and groupName is test

So, now eleven should be able to see eleven2 post if eleven2 share the post with the group

Let's find what is shared with:

```

mysql> select * from SharedWith;
+-----+-----+-----+
| groupOwner | groupName | photoID |
+-----+-----+-----+
| eleven2    | test     | 5 |
| eleven2    | test     | 6 |
| eleven2    | test     | 7 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

We can see here, there three shared photos, 5,6,7. From above picture we can know that eleven2 only post 6, 7. Which means that eleven also shared with group. But the query is not allowed photoid 5 since it is not posted by eleven2.

Let's see if eleven search eleven2

Homepage

Welcome Back, Zhengyi!

Here are your photos:

Photo	Time Posted
5	2019-12-05 10:12:52
8	2019-12-08 11:38:40
9	2019-12-08 11:38:50
10	2019-12-08 11:39:01
11	2019-12-08 11:39:14

[Upload a Photo](#)

[View All Visible Photos!](#)

[View a Profile](#)

[Follow Requests](#)

[Find Poster Picture](#)

[Logout](#)

Who you want to find

eleven2

This is pic post by your search poster

Photo	Time Posted
6	2019-12-07 19:44:39
7	2019-12-07 19:44:51

[Back](#)

Now we can see that when eleven search eleven2, it showed all the shared pictures with eleven2 posted! So, part 2 successes.

Now, with this two parts, query also need a OR statement that connect two rules. As following:

SELECT photoID, postingDate **FROM** Photo **WHERE** photoID **IN**

(part 1)

OR photoID **IN**

(part 2)

AND photoPoster = %s

If user search them self:

Who you want to find

eleven

This is pic post by your search poster

Photo	Time Posted
5	2019-12-05 10:12:52
8	2019-12-08 11:38:40
9	2019-12-08 11:38:50
10	2019-12-08 11:39:01
11	2019-12-08 11:39:14

[Back](#)

As we can see, eleven search itself, it can get all photo posted by eleven proved by above pictures.