Name:- Yash Rajendra Gaikwad
Data Analytics Trainee
Project 2:- Instagram User Analytics
Software Used:- My SQL Workbench 8.0 CE

TASK:-

A) Marketing Analysis:-

1) Loyal User Reward.
2) Inactive User Engagement.
3) Contest Winner Declaration.
4) Hashtag Research.
5) Ad Campaign Launch.

B) Investor Metrics:-

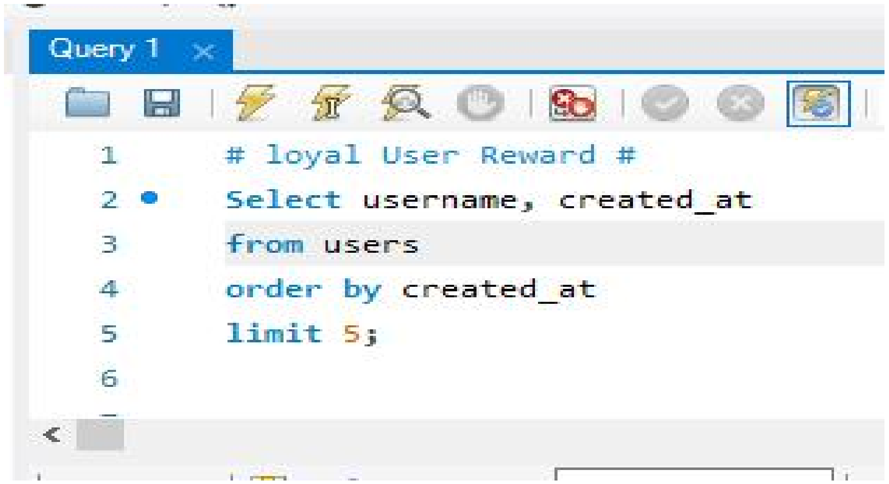
1) User Engagement.
2) Bots And Fake Accounts.

A) Marketing Analysis

1) Loyal User Reward:- Identify the five oldest users on instagram from the provided database.

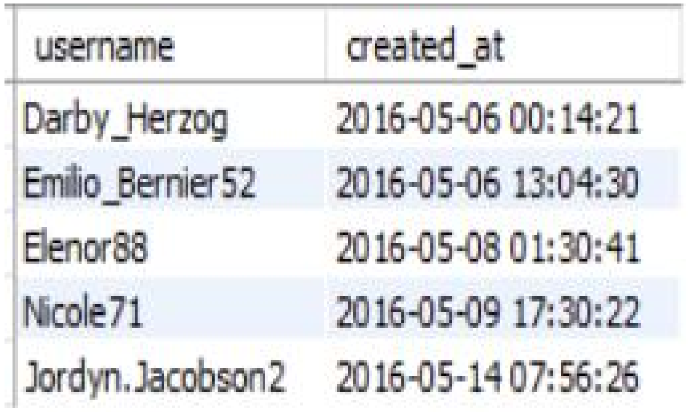
- Process:-
- First we Import the Database in to the MySQL.
- Now we use the data from the users table by selecting the username and created_at columns.
- Then we use the order by function so, we get order of the desired output by sorting it with the created_at column in ascending order.
- lastly we use the limit function, so that we will get only top 5 oldest Instagram users as a output.

➢ Query


Query 1

```
# loyal User Reward #
Select username, created_at
from users
order by created_at
limit 5;
```

➢ **Result:- Finaly, we get the Top 5 Most Loyal Instagram users.**

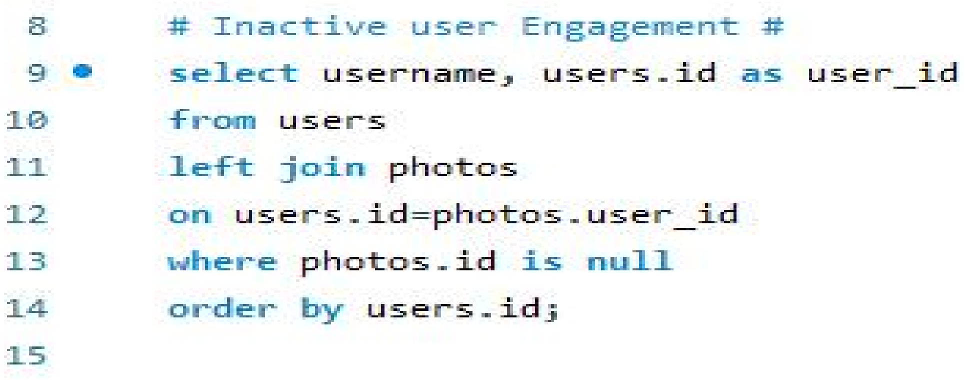


username	created_at
Darby_Herzog	2016-05-06 00:14:21
Emilio_Bernier52	2016-05-06 13:04:30
Elenor88	2016-05-08 01:30:41
Nicole71	2016-05-09 17:30:22
Jordyn.Jacobson2	2016-05-14 07:56:26

2) Inactive Users Engagement:- Identify the users who have never posted a single photo on Instagram.

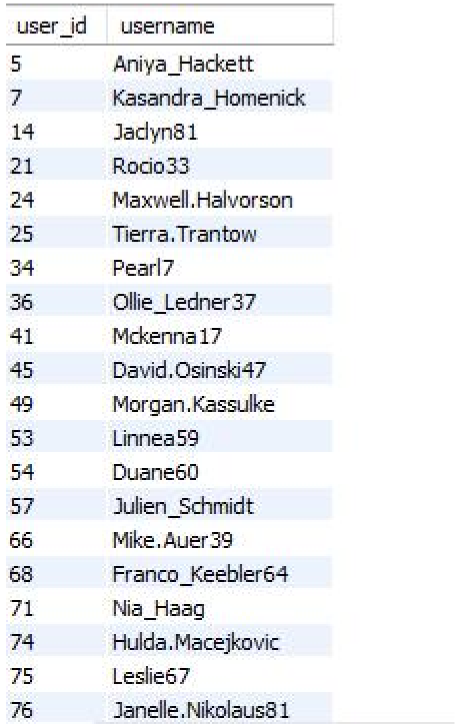
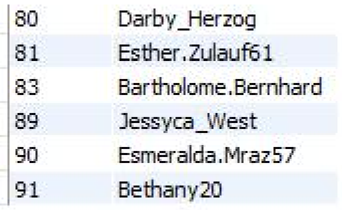
- Process:-
- We will first select username, users.id and user_id column from the users table.
- Then we will left join photos table on the users table, on users.id = photos.user_id.
- We use it because, both the users.id and photos.user_id have common contents in them.
- Then we will find rows from the users table where the photos.id is null.
- Lastly we use order by usier.id function and we will gwt the users who have never posted a single photo on Instagram.

- Query:- .



```
# Inactive user Engagement #
select username, users.id as user_id
from users
left join photos
on users.id=photos.user_id
where photos.id is null
order by users.id;
```

- **Result:-** **There are total 26 Instagram users who have never posted a single photo on Instagram.**

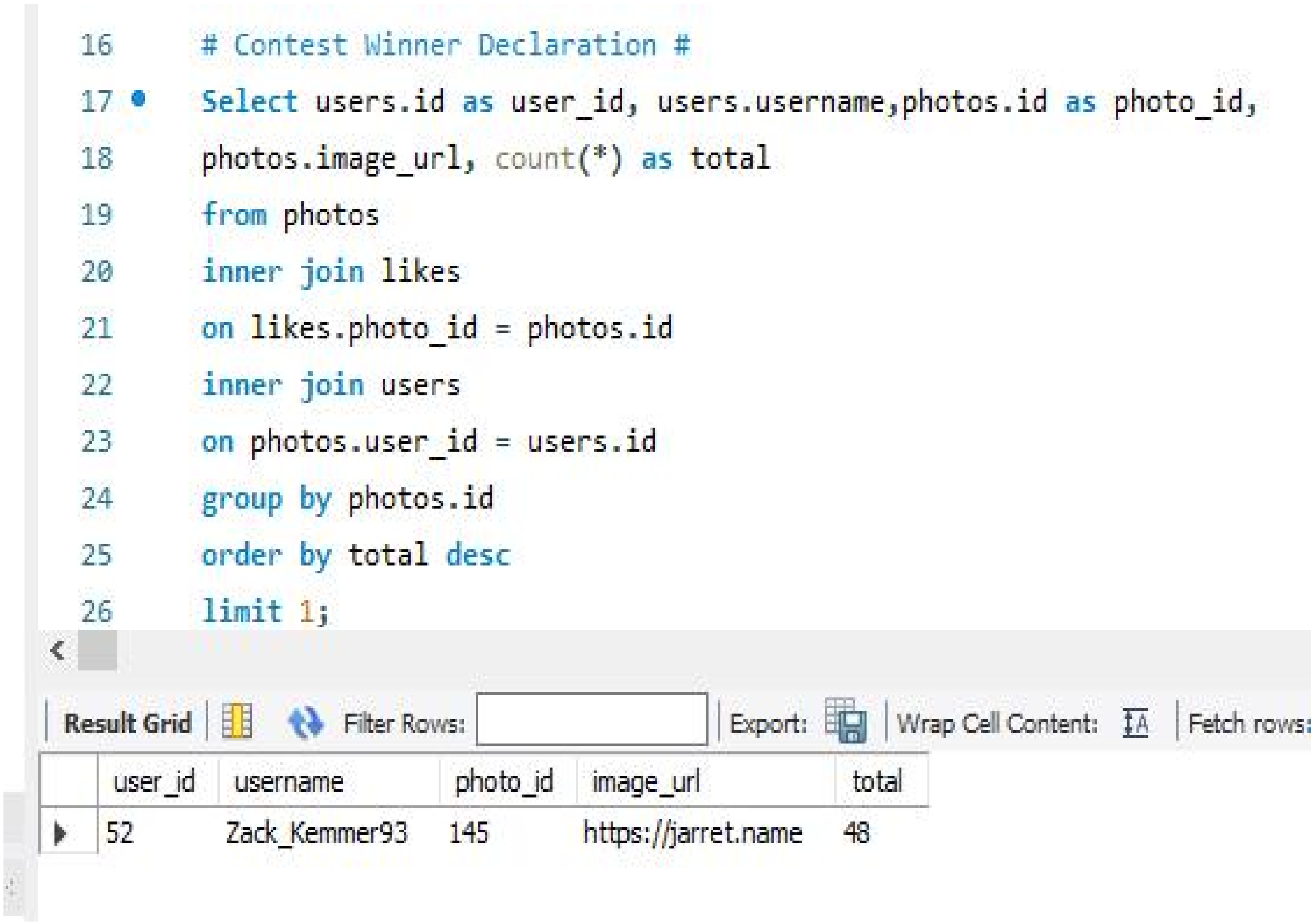



user_id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

3) Contest Winner Declaration:- Determine the winner of the contest and provide their details to the team.

- **Process:-**
- First we will select the users.id as user_id, users.username, photos.id as photo_id, photos.image_url and count(*) as total from photo.
- 2. Then, we will use the inner join the three tables wiz : photos, likes and users, on likes.photo_id = photos.id and photos.user_id = users.id.
- 3. Then, we use the group by function so we will get the output on the basis of photos.id
- 4. After that we use order by function so we will get the data on the basis of the total in descending order
- 5. Finaly, we will using limit function to view only the top liked photo's information

➢ **Query And Result:-** **After Analysis, we find that Zack_Kemmer93 is winner of Contest.**



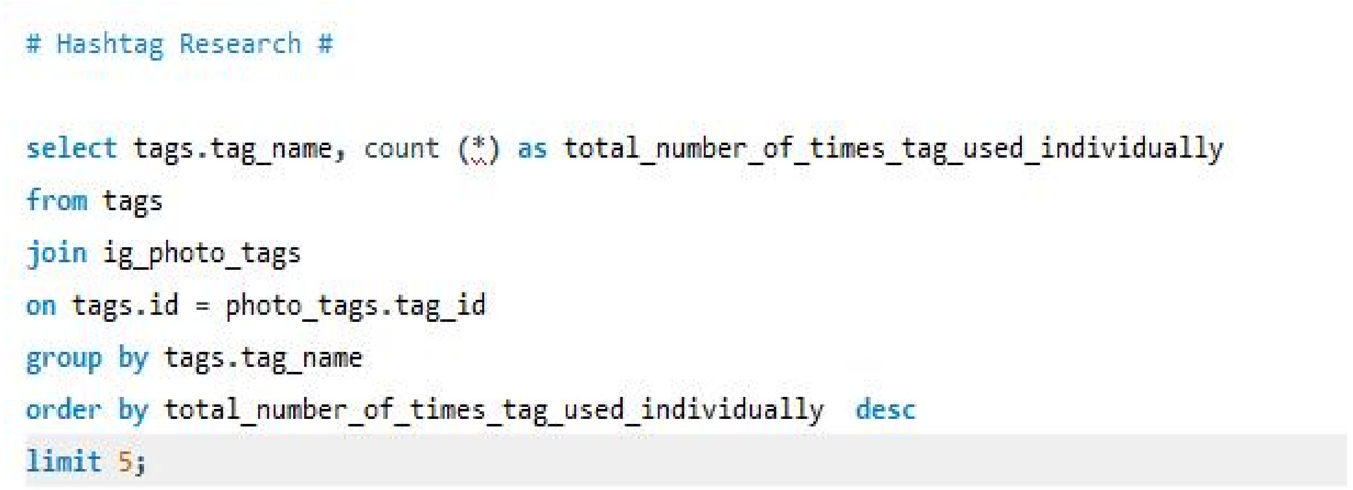
```
# Contest Winner Declaration #
Select users.id as user_id, users.username,photos.id as photo_id,
photos.image_url, count(*) as total
from photos
inner join likes
on likes.photo_id = photos.id
inner join users
on photos.user_id = users.id
group by photos.id
order by total desc
limit 1;
```

user_id	username	photo_id	image_url	total
52	Zack_Kemmer93	145	https://jarret.name	48

4) Hashtag Research:- Identify and suggest the top five most commonly used hashtags on the platform.

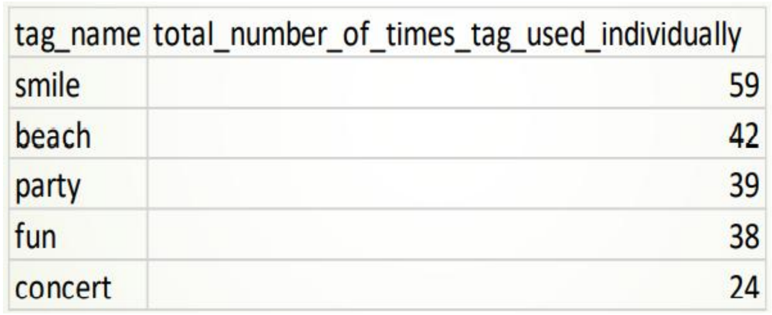
- **Process:-**
- To find the top 5 most commonly used hashtags on Instagram first we need to select the tag_name column from the tag table.
- We use the count(*) as total_number_of_times_tag_used_individually so as to count the number of tags used individually.
- Then, we use the join function for tags table and photo_tags table, on tags.id = photo_tags.tag_id cause they contain the same content in them i.e. tag_id
- After that we use group by function to get the desired output on the basis of tags.tag_name
- Then using the order by function we need to sort the output on the basis of total(total number of tags per tag_name) in descending order
- Finally, to find the top 5 most used tag names we will use the limit 5 function.

- **Query:-**



```
# Hashtag Research #

select tags.tag_name, count (*) as total_number_of_times_tag_used_individually
from tags
join ig_photo_tags
on tags.id = photo_tags.tag_id
group by tags.tag_name
order by total_number_of_times_tag_used_individually  desc
limit 5;
```

- **Result:- After anakysis we find that the Smile is the most used Hashtag followed by beach, party, fun, concert.**

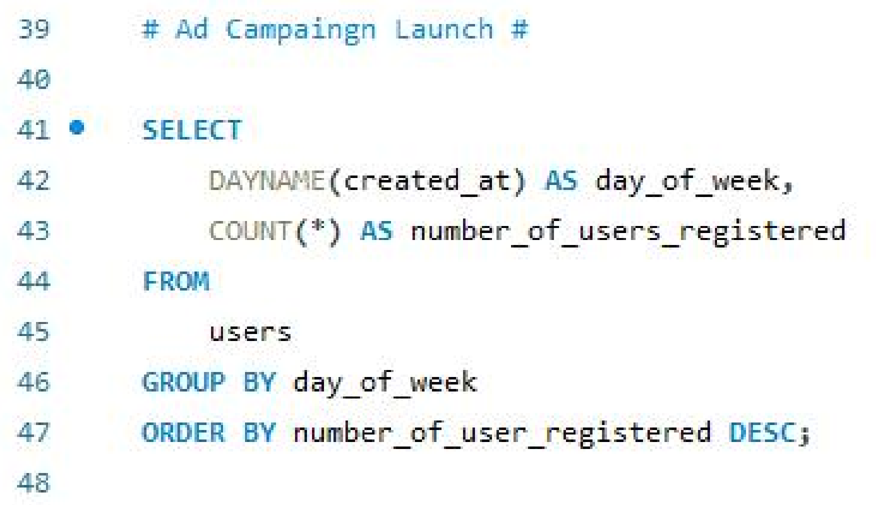


tag_name	total_number_of_times_tag_used_individually
smile	59
beach	42
party	39
fun	38
concert	24

5) Ad Campaign Launch:- Determine the day of the week when most users register on instagram. Provide insights on when to schedule an ad campaing.

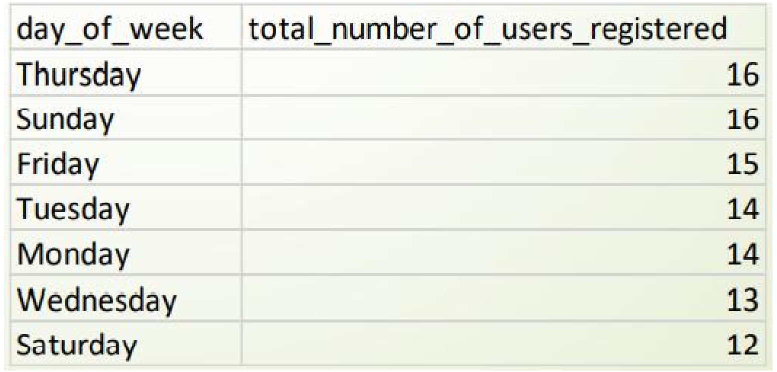
- **Process:-**
- To launch the Ad Campaign we have to find the day of week on which most users register on Instagram.
- First we need to define the columns of the desired output table using select dayname(created_at) as day_of_week.
- Same we use for the count(*) as total_number_of_users_registered from the users table
- Then we use the group by function so we get the output table on the basis of day_of_week.
- Lastly we use the order by function we sort the output table on the basis of total_number_of_users_registered in descending order.

- **Query:-**



```
# Ad Campaingn Launch #

SELECT
    DAYNAME(created_at) AS day_of_week,
    COUNT(*) AS number_of_users_registered
FROM
    users
GROUP BY day_of_week
ORDER BY number_of_user_registered DESC;
```

- **Result:-** **After the analysis we find that most of the users registered on Thursday and Snday. So, it would beneficial to start Ad on Thursday and Saturday,**



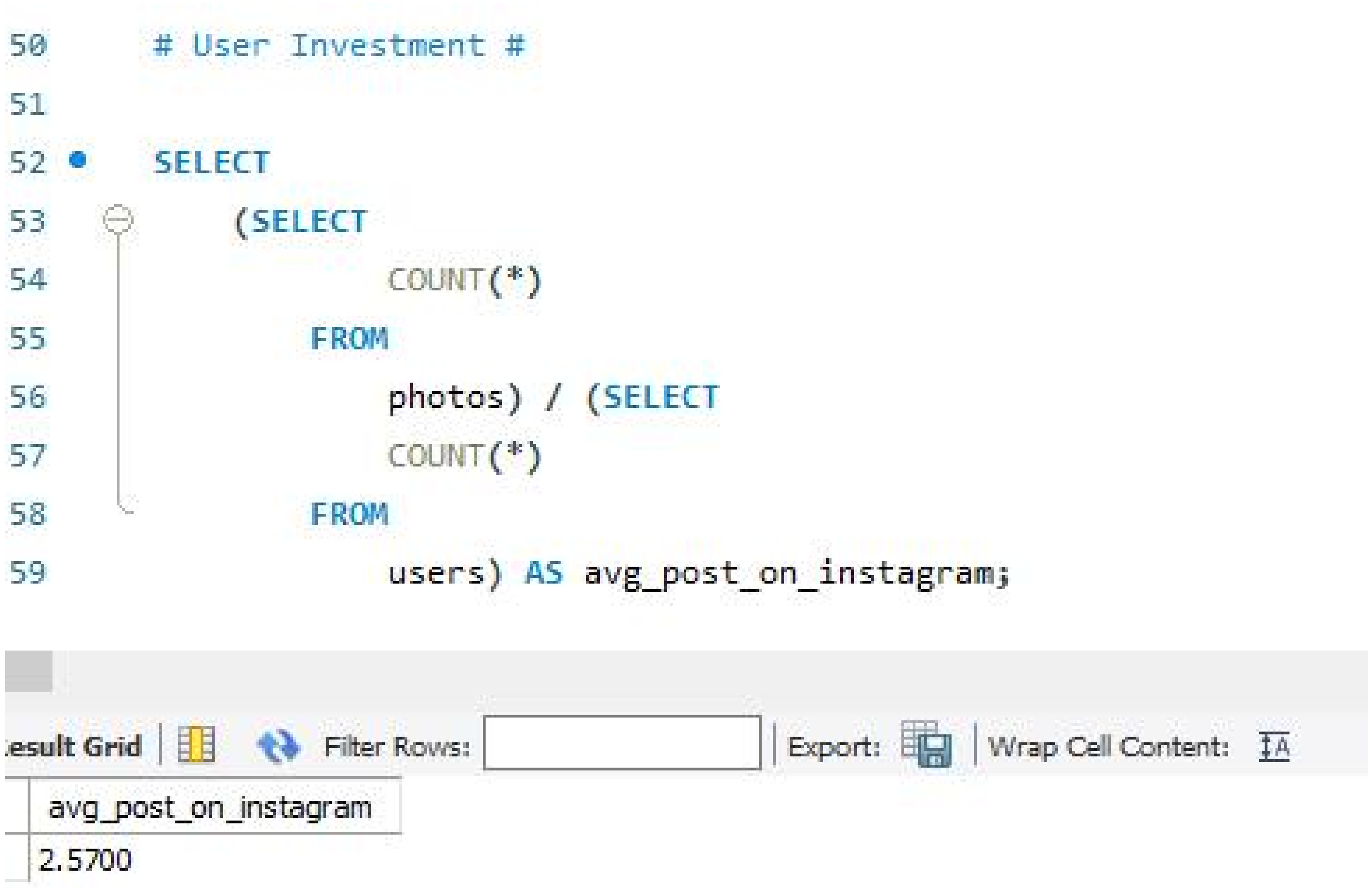
day_of_week	total_number_of_users_registered
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

B) Investor Metrics

1) User Engagement:- Calculate the average number of posts per user on instagram. Also, provide the total number of photos on instagram divided by the total number of users.

- **Process:-**
- In this we need to perform two operations
- **First Case**
- First, we need to find first the count number of photos(posts) that are present in the photos.id column of the photos table i.e. count(*) from photos.
- Similarly, we need to find the number of users that are present in the users.id column of the users table i.e. count(*) from users.
- We divide both the values count(*) from photos/count(*) from users and hence we would get the total number of photos / total number of users.
- Lastly how many times the users posts on Instagram we need to find the total occurrences of each user_id in photos table

- **Quary And Result For First Case:-** After the analysis we found out that the average user posts around 2.57 times on the Instagram.



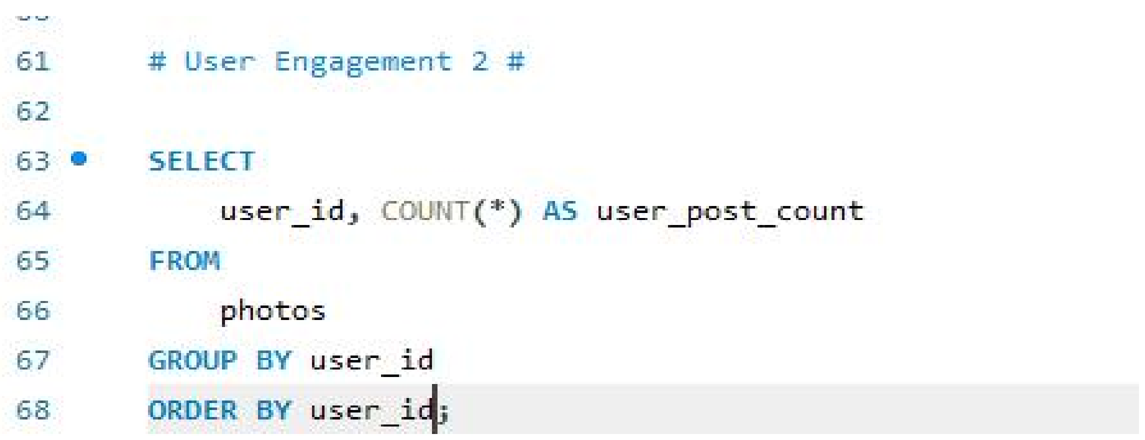
```
# User Investment #

SELECT
    (SELECT
            COUNT(*)
        FROM
            photos) / (SELECT
            COUNT(*)
        FROM
            users) AS avg_post_on_instagram;
```

```
avg_post_on_instagram
2.5700
```

Second Case

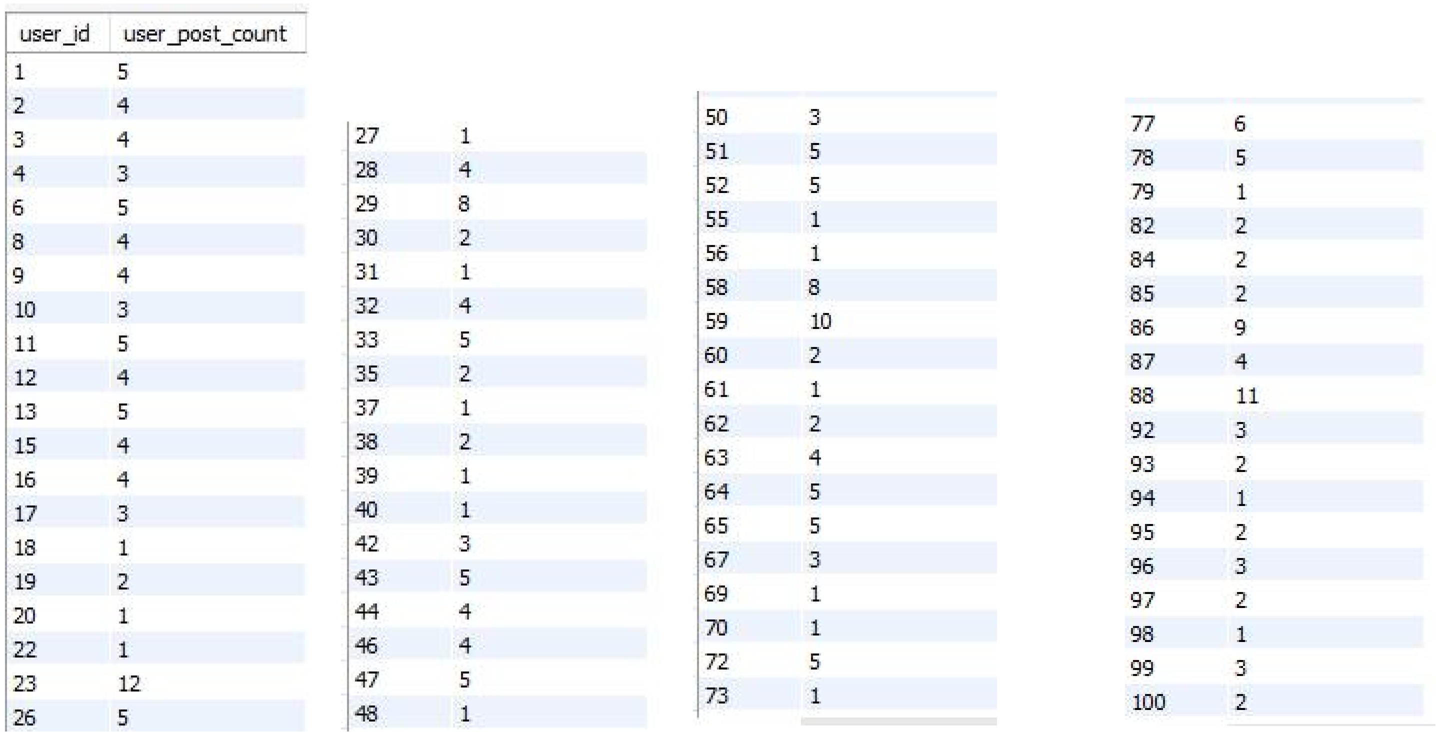
- In second case we need to find the total number of photos on Instagram OR the total number of users.
- First we Select the user_id also we use the count(*) function to find the number of photos on Instagram OR the total number of users and name it as user_post_count from the photos
- Then, we use the group by function to get the output table according to the user_id.
- Lastly, we use the order by function to sort the output tabe by user_id

Query:-



```
# User Engagement 2 #

SELECT
    user_id, COUNT(*) AS user_post_count
FROM
    photos
GROUP BY user_id
ORDER BY user_id;
```

- **Result:-** This is the list of Total number of photos on instagram OR the total number of users on Instagram.

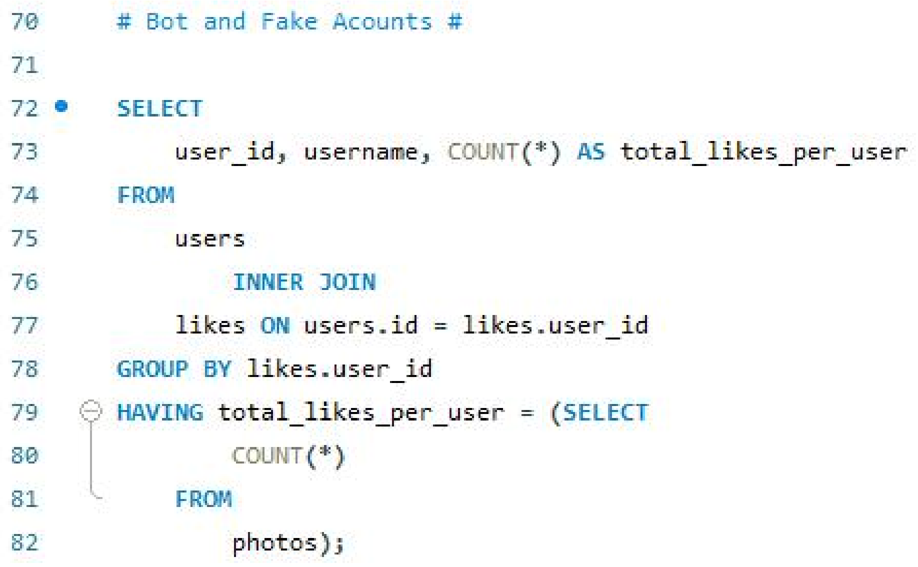


user_id	user_post_count
1	5
2	4
3	4
4	3
6	5
8	4
9	4
10	3
11	5
12	4
13	5
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	12
26	5
27	1
28	4
29	8
30	2
31	1
32	4
33	5
35	2
37	1
38	2
39	1
40	1
42	3
43	5
44	4
46	4
47	5
48	1
50	3
51	5
52	5
55	1
56	1
58	8
59	10
60	2
61	1
62	2
63	4
64	5
65	5
67	3
69	1
70	1
72	5
73	1
77	6
78	5
79	1
82	2
84	2
85	2
86	9
87	4
88	11
92	3
93	2
94	1
95	2
96	3
97	2
98	1
99	3
100	2

2) Bot And Fake Accounts:- Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

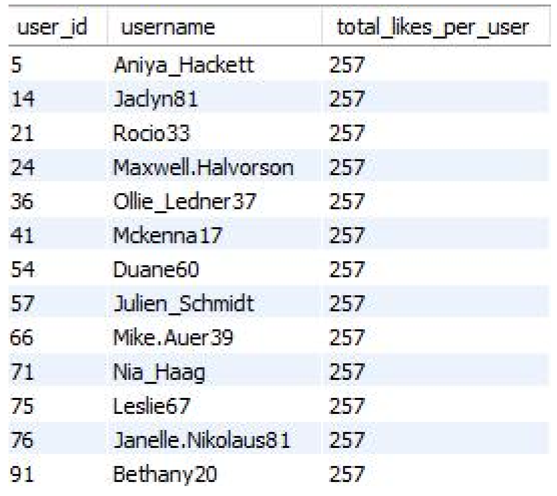
- **Process:-**
- To find the bots and fake accounts First, we select the user_id column from the photos table and the username column from the users table
- Then, we use the count(*) function to count the total number of likes from the likes table.
- After that we use the inner join function on users and likes table on the basis of users.id and likes.user_id.
- Now we use the group by function so that we get the output table on the basis of likes.user_id.
- Lastly, we search for the values from the cout(*) from photos having equal values with the total_likes_per_user.

- **Query:-**



```
# Bot and Fake Acounts #

SELECT
    user_id, username, COUNT(*) AS total_likes_per_user
FROM
    users
        INNER JOIN
    likes ON users.id = likes.user_id
GROUP BY likes.user_id
HAVING total_likes_per_user = (SELECT
        COUNT(*)
    FROM
        photos);
```

- **Result:- After analysis we find the Bots and Fake acounts.**



user_id	username	total_likes_per_user
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

END