Instagram User Analytics

By :- Pritam Dey

Analysis done on following topics:

1. **Marketing :-**

- **Rewarding Most Loyal Users**
- **Remind Inactive Users to Start Posting**
- **Declaring Contest Winner**
- **Hashtag Researching**
- **Launch AD Campaign**

1. **Investor Metrics :-**

- **User Engagement**
- **Bots & Fake Accounts**

SOFTWARE USED :- **MySQL Workbench 8.0 CE**

Marketing

Rewarding the most loyal user :- People who have been using the platform for the longest time (Top 5 oldest Instagram user.

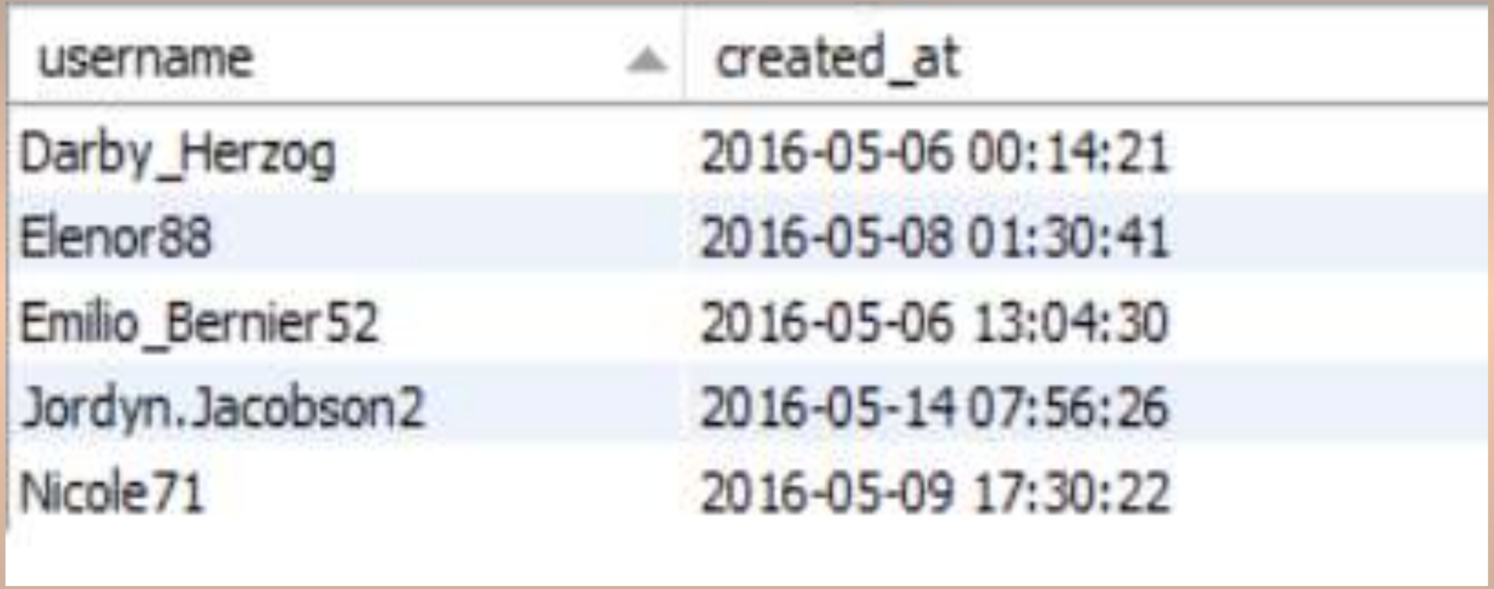
STEPS :-

1. We will use the data from “user” table by selecting the “username” and “created_at” columns
2. Then using the “order by” function we will order the desire output by sorting the “created_at” column in Ascending order.
3. Using the “limit” function to limited the output in our desired limit.

QUERY :-

```
Select username, created_at from users order by created_at ASC limit 5;
```

Output:



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

Remind Inactive Users to start Posting : Remind inactive users to start posting (users who never post a single photo in the Instagram.)

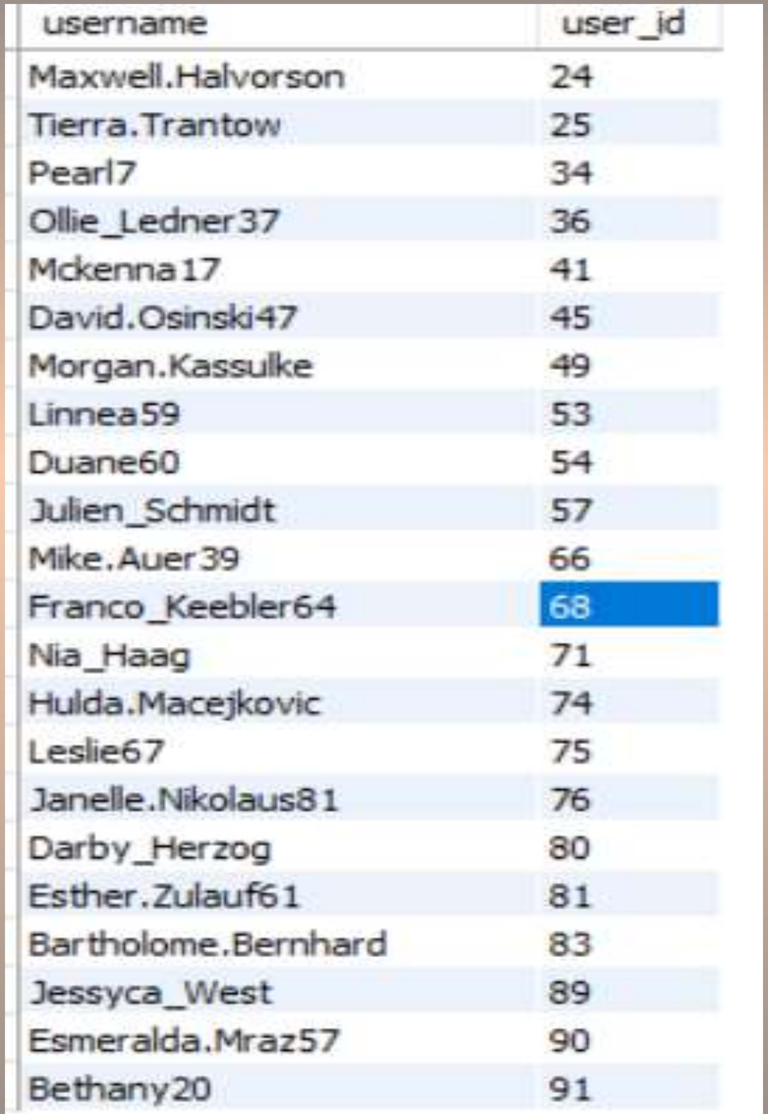
STEPS:

1. We will first select the “username” column from the “users” table.
2. Then we will LEFT JOIN “Photos” table on the “users” table on “users.id=photos.user_id” both the users.id and photos.user_id have the common contents in them.
3. Then just find the rows from the “users” table where “phots.id =NULL”.

QUERY :-

```
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;
```

OUTPUT



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

Declaring contest winner : The team started a contest and the user who will get the most like in a single picture will win the contest . Now the team wants to declare the winner of the contest. Identify the winner of the contest and provide details of that user.

STEPS :

1. **First we will select the “users.username, photos.id, photos.image_url “and count(*) as total.**

2. Then, we will “INNER JOIN” the three tables: photos, likes and users, on “likes.photo_id = photos.id and photos.user_id = users.id”

3. Then, by using “GROUP BY” function we will group the output on the basis of “photos.id”

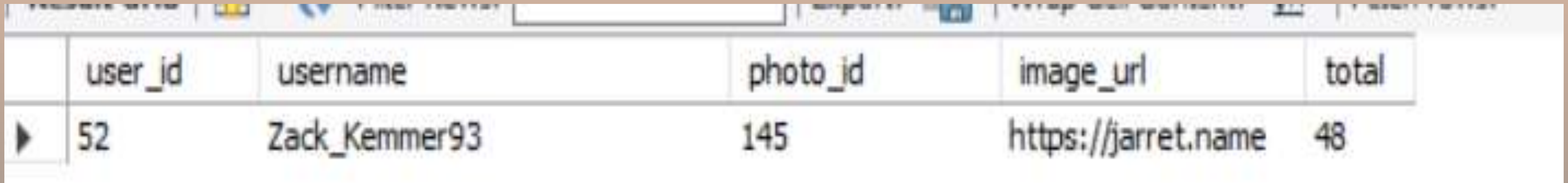
4. Then, using order by function we will sorting the data on the basis of the total in “DESCENDING ORDER”

5. Then, to find the most liked photo we will using “LIMIT” function to view only the top liked photo’s information .

QUERY :

```
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;
```

OUTPUT:



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

Hashtag Researching : A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.(Top 5 commonly used #Hashtags on Instagram)

STEPS:

1. We need to SELECT the “tag_name” column from the “tagS” table and the “count(*) as total” function so as to count the number of tags used individually.

2. Then, we need to JOIN “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

3. Then using the GROUP BY function we need to group the desired output on the basis of “tags.tag_name”

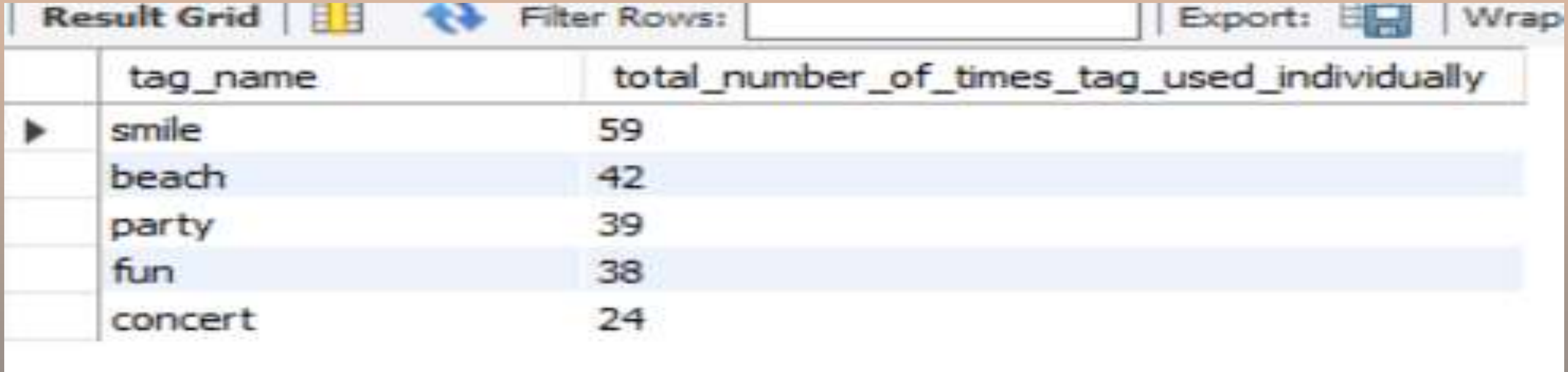
4. Then using the ORDER BY function we need to sort the output on the basis of total(total number of tags per tag_name) in DESCENDING ORDER

5. Then, to find the top 5 most used tag names we will use the limit 5 function.

QUERY :

```
select tags.tag_name, count(*) as total_number_of_times_tag_used_individually
from tags
join 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;
```

OUTPUT:



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

Launch AD Campaign : The team wants to know, which day would be the best day to launch ADs. (What day of the week do most users register on?)

STEPS:

1. First we define the columns of the desired output table using **SELECT dayname(created_at) as day_of_week and count(*) as total_number_of_users_registered** from the **users** table

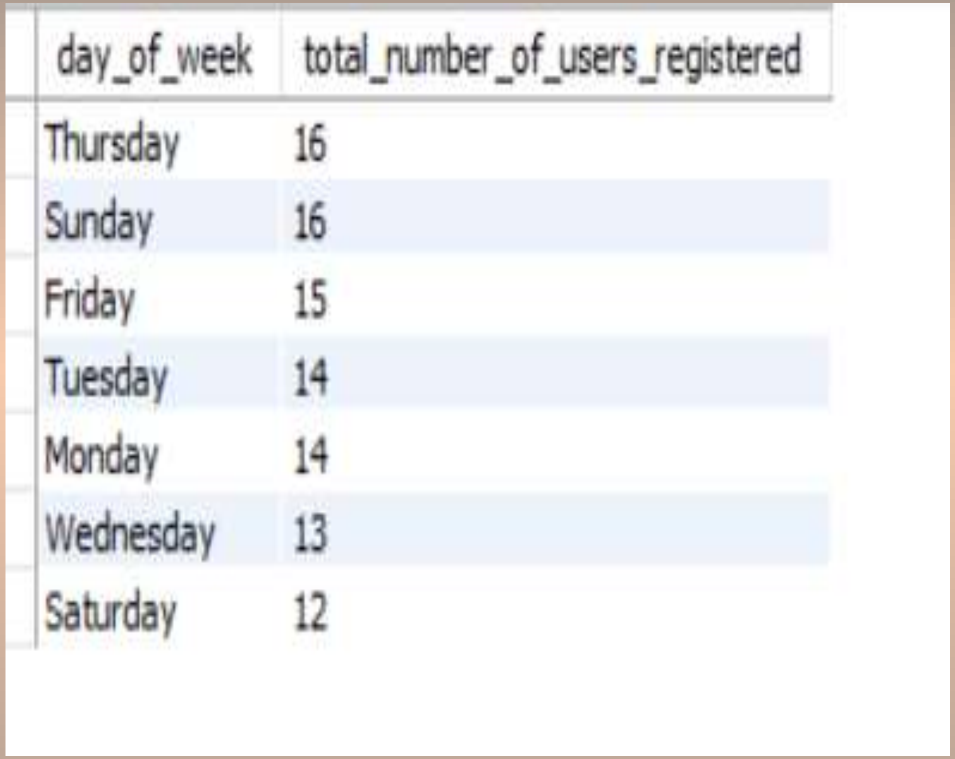
2. Then using the **GROUP BY** function we group the output table on the basis of **day_of_week**

3. Then using the **order by** function we order/sort the output table on the basis of **total_number_of_users_registered** in **descending order**

QUERY:

```
select dayname(created_at) as day_of_week, count(*) as total_number_of_users_registered
from users group by day_of_week order by total_number_of_users_registered DESC;
```

OUTPUT:



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

Most of the users registered on Thursday and Sunday i.e. 16 and hence it would prove beneficial to start AD Campaign on these two days

INVESTOR METRICS

User Engagement : Are users still as active and post on Instagram or they are making fewer posts. How many times does average user posts on Instagram? Also, provide the total number of photos on Instagram/total number of users.

STEPS:

1. 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**
2. 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**

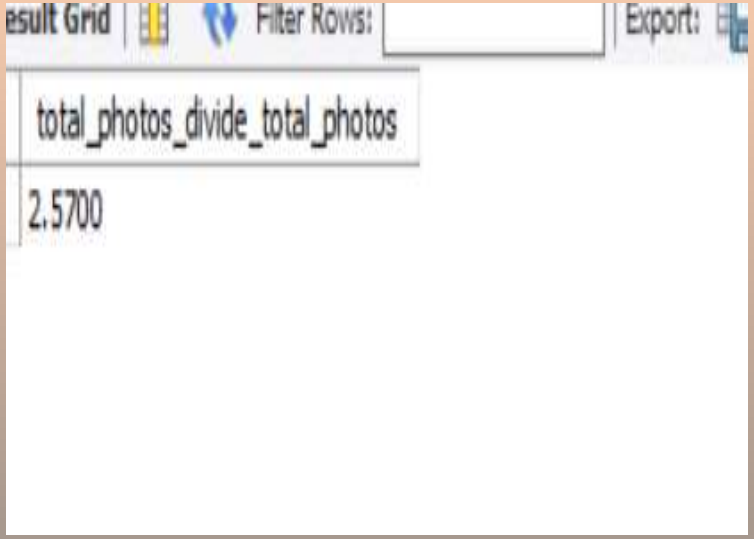
3. Next, we need to divide both the values i.e. **count(*) from photos/count(*) from users** and hence we would get the total number of photos / total number of users

4. To find how many times the users posts on Instagram we need to find the total occurrences of each **user_id** in photos table

QUERY:

```
select(select count(*) from photos)/(select count(*) from users) as
total_photos_divide_total_photos;
```

OUTPUT:



```
total_photos_divide_total_photos
2.5700
```

So, there are in total 257 rows i.e. 257 photos in the photos table and 100 rows i.e. 100 ids in the users table which makes the desired output to be 257/100 = 2.57

User Engagement : **Are users still as active and post on Instagram or they are making fewer posts. How many times does average user posts on Instagram? Also, provide the total number of photos on Instagram/total number of**

STEPS:

1.SELECT the **user_id , count(*) as user_post_count** from **photos** table.

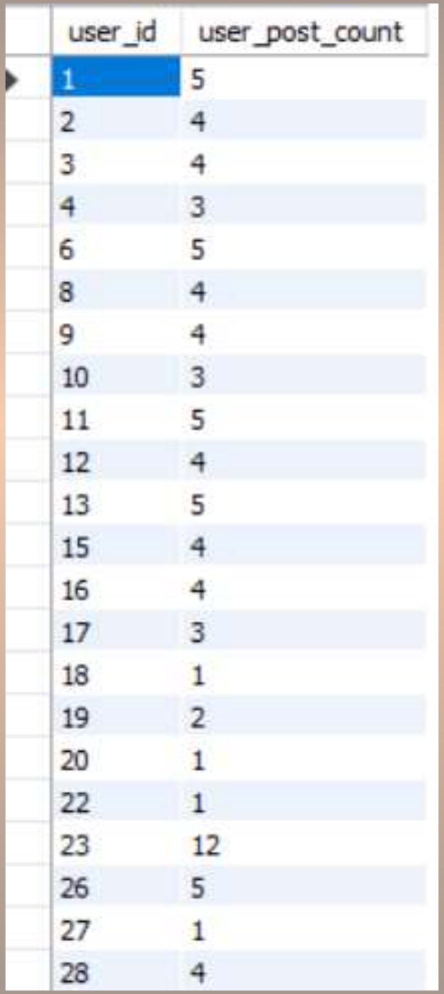
2. Then use the **GROUP BY** function to group the **user_id**.

3. Then use the **ORDER BY** function to **user_id** for desired output.

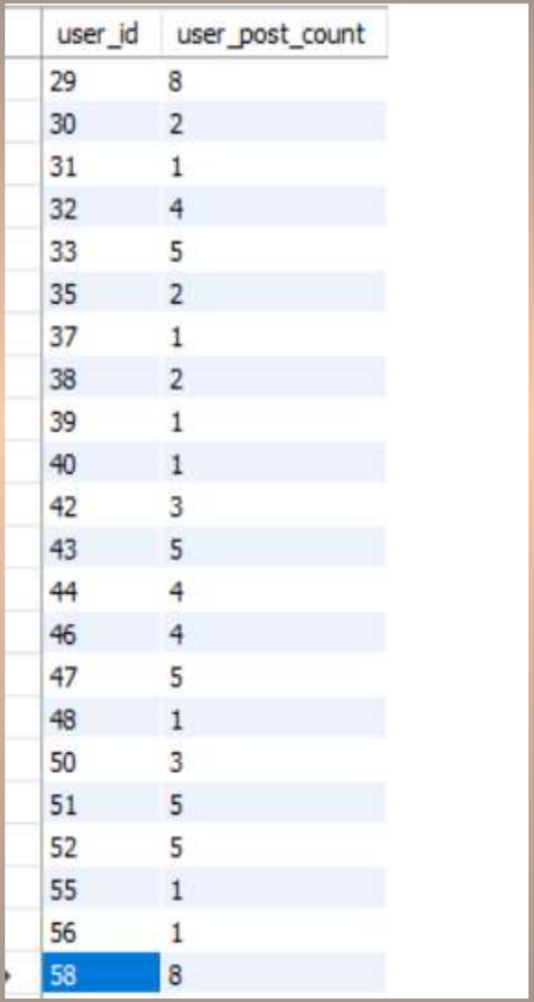
QUERY:

```
select user_id,count(*) as user_post_count
from photos
group by user_id
order by user_id;
```

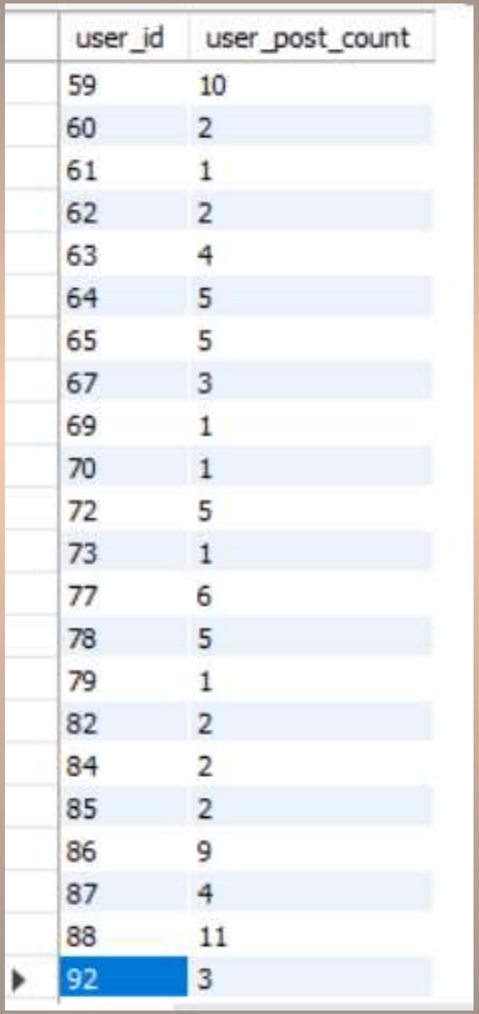
OUTPUT:



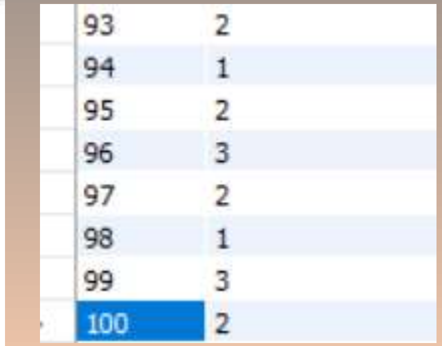
user_id	user_post_count
1	5
2	4
3	4
4	3
6	5
8	4
9	4
10	3
11	5
12	4
13	5
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	12
26	5
27	1
28	4



user_id	user_post_count
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



user_id	user_post_count
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

So the user_id along with the number of times each user_id has posted is provided.

Bots and Fake Accounts : The investors want to know if the platform is crowded with fake and dummy accounts. Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).
To find the bots and fake accounts :

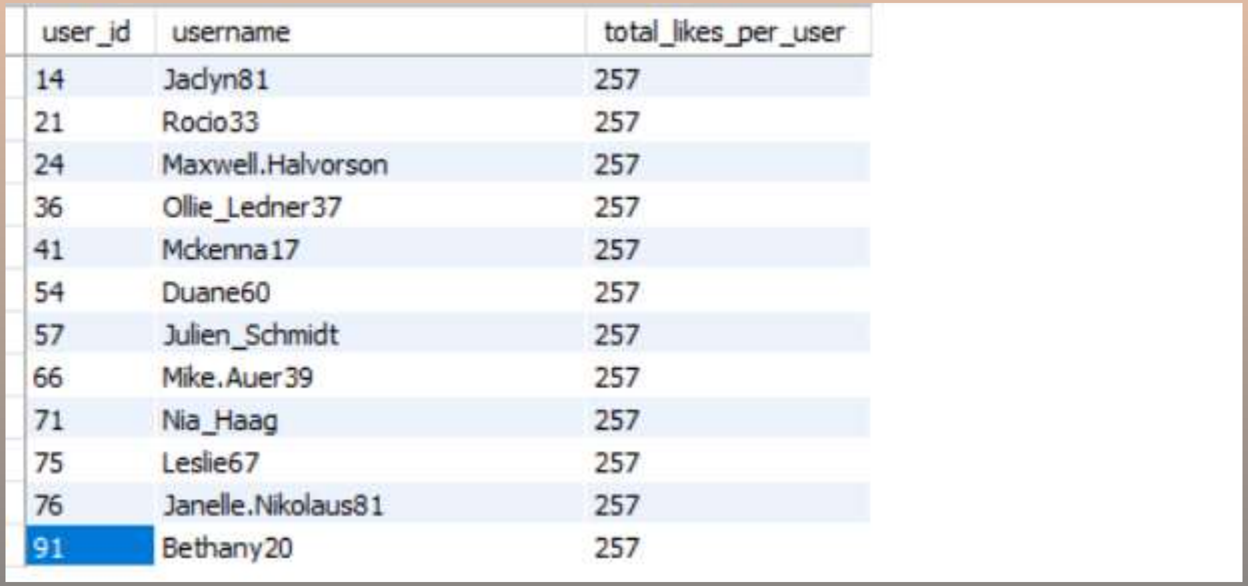
STEPS:

1. First, we select the user_id column from the photos table
2. Then we select the username column from the users table
3. Then, we select the count(*) function to count total number of likes from the likes table
4. Then we inner join users and likes table on the basis of users.id and likes.user_id, using the on function/clause
5. Then by using the group by function we group the desired output table on the basis of likes.user_id
6. Then, we search for the values from the cout(*) from photos having equal values with the total_likes_per_user

QUERY:

```
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);
```

OUTPUT:



user_id	username	total_likes_per_user
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

So, the users along with their respective username, user_id and total_likes_per_user have been provided. This user_ids may be bots or fake accounts

THANK YOU