TRAINITY PROJECT
TASK 2

Instagram User Analytics

PRESENTED BY ADITYA PALANDE

email: adityap.works@gmail.com

PART A

Marketing

[software used: MySQL workbench 8.0 ce]

Loyal User Reward:

The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

Task: Identify the five oldest users on Instagram from the provided database.

APPROACH:

1.Select `users` table.

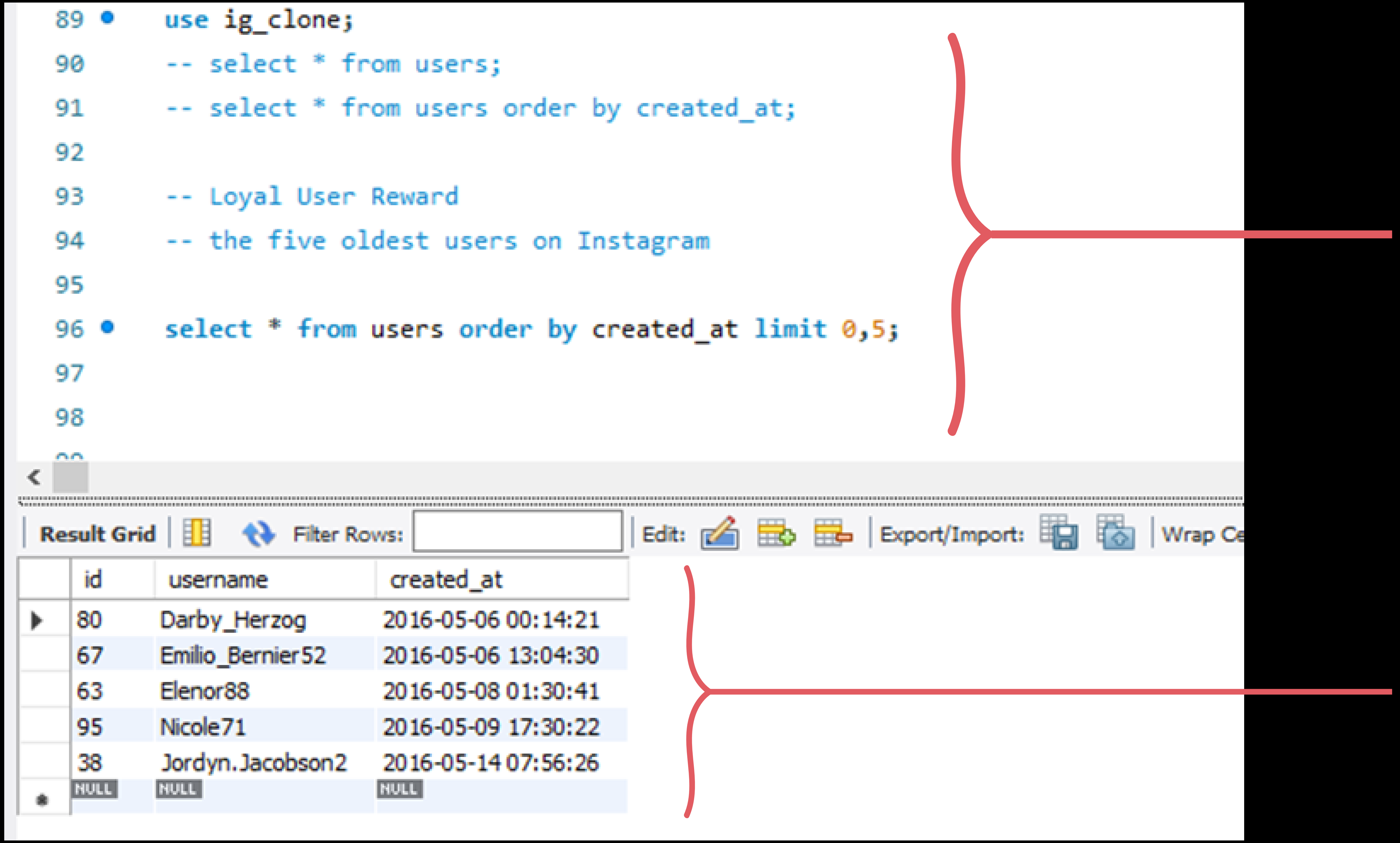
```
select * from users...
```

2.Display rows in ascending order based on `created_at` column.

```
...order by created_at...
```

3.Display only the first five entries to get the five oldest users on instagram.

```
...limit 5;
```



```sql
use ig_clone;
-- select * from users;
-- select * from users order by created_at;

-- Loyal User Reward
-- the five oldest users on Instagram

select * from users order by created_at limit 0,5;
```

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

Query

Five oldest users on instagram

Inactive User Engagement:

The team wants to encourage inactive users to start posting by sending them promotional emails.

Task: Identify users who have never posted a single photo on Instagram.

APPROACH:

1.Select `users` table.

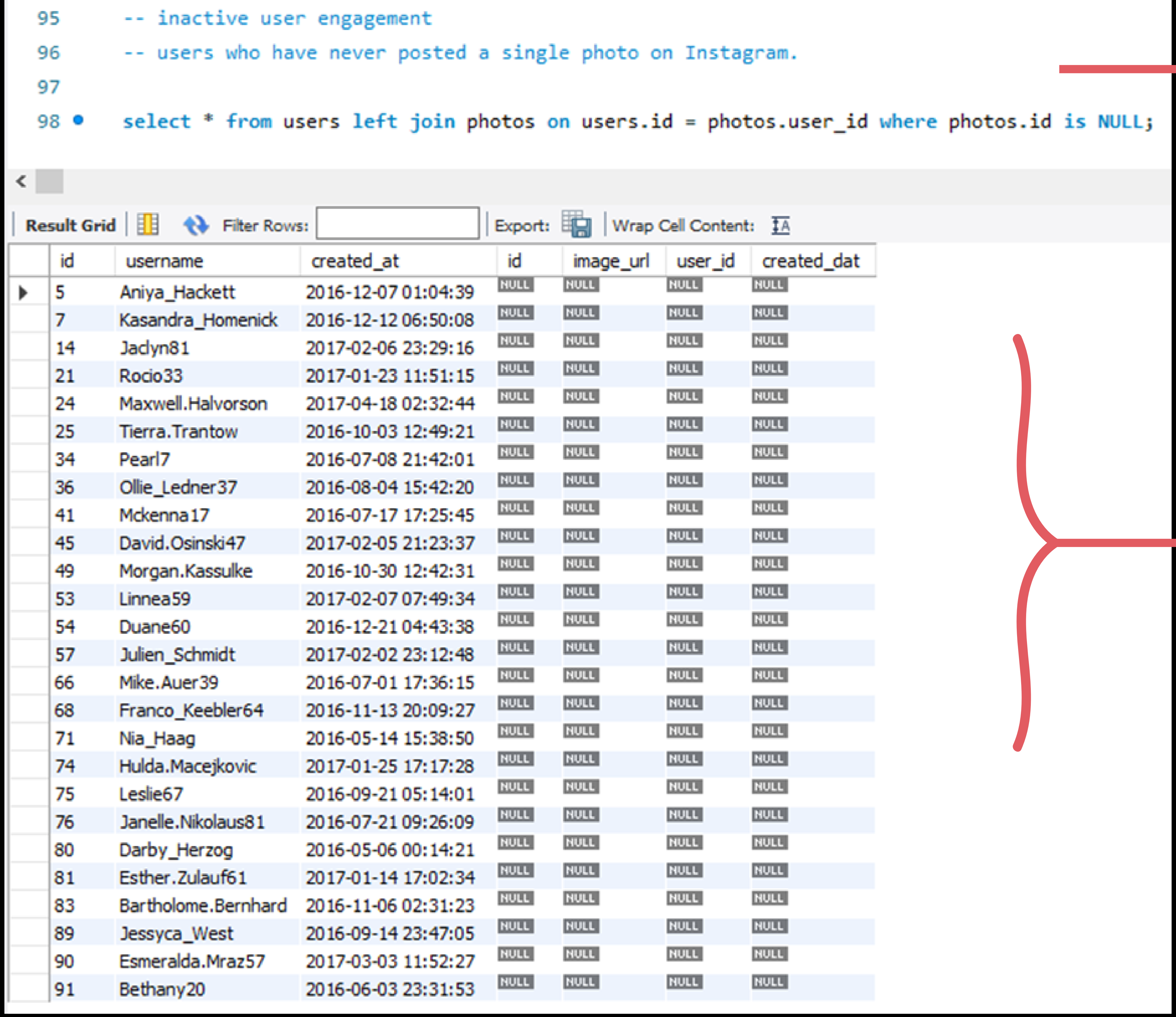
```
      select * from users...
```

2.left join users table and photos table on users.id = photos.user_id

```
      ...left join photos on users.id = photos.user_id...
```

3.Use where clause to filter on those rows that are NULL.

```
      ...where photos.id is NULL;
```



```
-- inactive user engagement
-- users who have never posted a single photo on Instagram.

select * from users left join photos on users.id = photos.user_id where photos.id is NULL;
```

id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jaclyn81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
21	Rocio33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
24	Maxwell.Halvorson	2017-04-18 02:32:44	NULL	NULL	NULL	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	Mckenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL
45	David.Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
49	Morgan.Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL
54	Duane60	2016-12-21 04:43:38	NULL	NULL	NULL	NULL
57	Julien_Schmidt	2017-02-02 23:12:48	NULL	NULL	NULL	NULL
66	Mike.Auer39	2016-07-01 17:36:15	NULL	NULL	NULL	NULL
68	Franco_Keebler64	2016-11-13 20:09:27	NULL	NULL	NULL	NULL
71	Nia_Haag	2016-05-14 15:38:50	NULL	NULL	NULL	NULL
74	Hulda.Macejkovic	2017-01-25 17:17:28	NULL	NULL	NULL	NULL
75	Leslie67	2016-09-21 05:14:01	NULL	NULL	NULL	NULL
76	Janelle.Nikolaus81	2016-07-21 09:26:09	NULL	NULL	NULL	NULL
80	Darby_Herzog	2016-05-06 00:14:21	NULL	NULL	NULL	NULL
81	Esther.Zulauf61	2017-01-14 17:02:34	NULL	NULL	NULL	NULL
83	Bartholome.Bernhard	2016-11-06 02:31:23	NULL	NULL	NULL	NULL
89	Jessyca_West	2016-09-14 23:47:05	NULL	NULL	NULL	NULL
90	Esmeralda.Mraz57	2017-03-03 11:52:27	NULL	NULL	NULL	NULL
91	Bethany20	2016-06-03 23:31:53	NULL	NULL	NULL	NULL

Query

Inactive users on instagram

Contest Winner Declaration:

The team has organized a contest where the user with the most likes on a single photo wins.

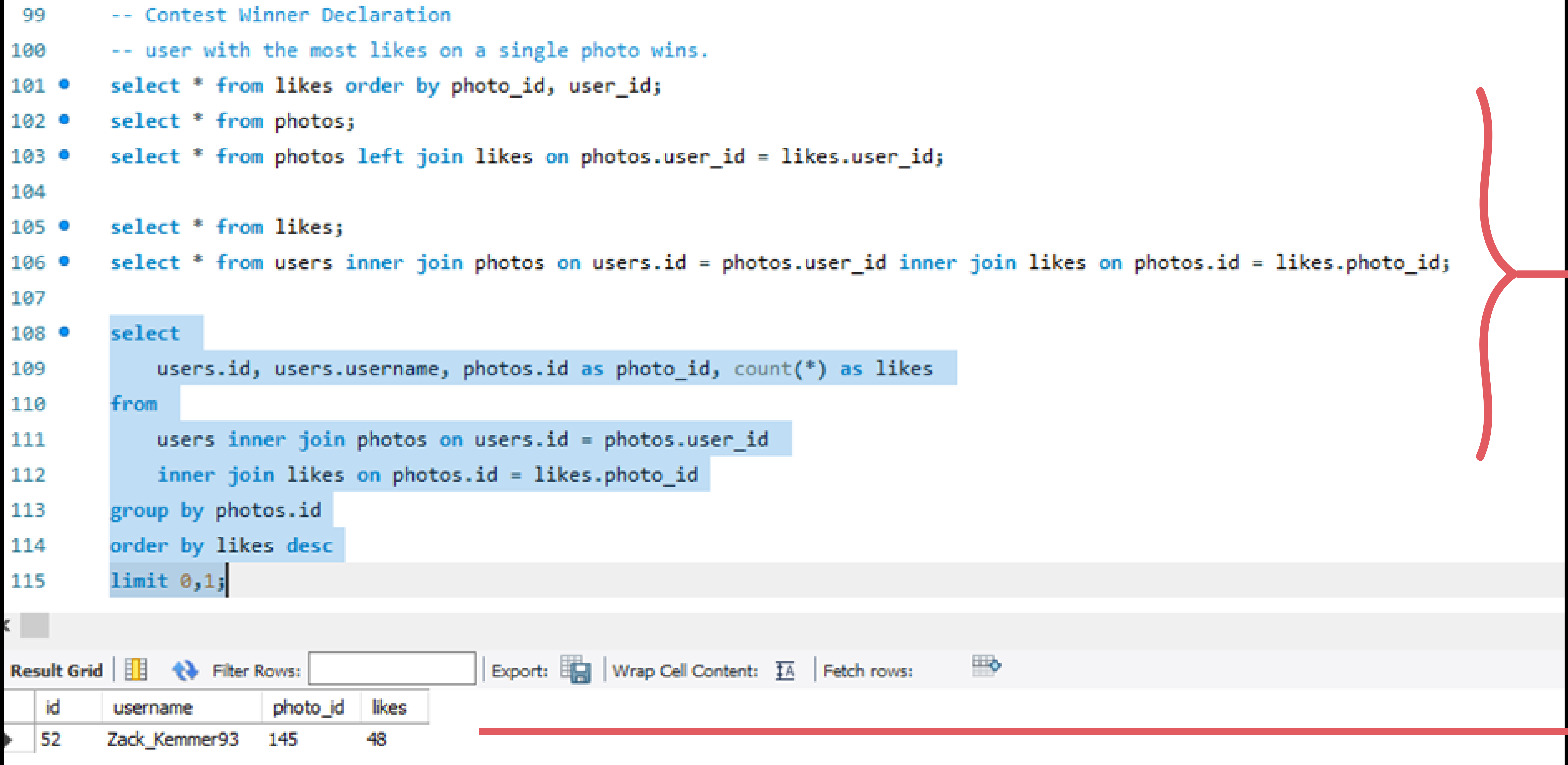
Task: Determine the winner of the contest and provide their details to the team.

APPROACH:

1.Select users.id, users.username, photos.id as photo_id,and count(*) as likes from `users` table.

2.inner join users, photos, likes tables on photos.id = likes.photo_id and photos.user_id = users.id

3.Group by photo_id and arrange the rows in descending order based on likes and finally select the first entry using limit.



```sql
-- Contest Winner Declaration
-- user with the most likes on a single photo wins.
select * from likes order by photo_id, user_id;
select * from photos;
select * from photos left join likes on photos.user_id = likes.user_id;

select * from likes;
select * from users inner join photos on users.id = photos.user_id inner join likes on photos.id = likes.photo_id;

select
    users.id, users.username, photos.id as photo_id, count(*) as likes
from
    users inner join photos on users.id = photos.user_id
    inner join likes on photos.id = likes.photo_id
group by photos.id
order by likes desc
limit 0,1;
```

id	username	photo_id	likes
52	Zack_Kemmer93	145	48

Query

Winner

Hashtag Research:

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

Task: Identify and suggest the top five most commonly used hashtags on the platform.

APPROACH:

1.Select tag_name and count(*) as `used` from `tags` table.

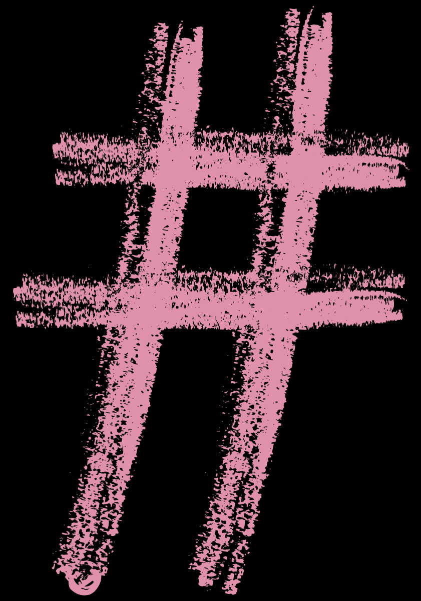
```
      select tag.tag_name, count(*) as used from tags...
```

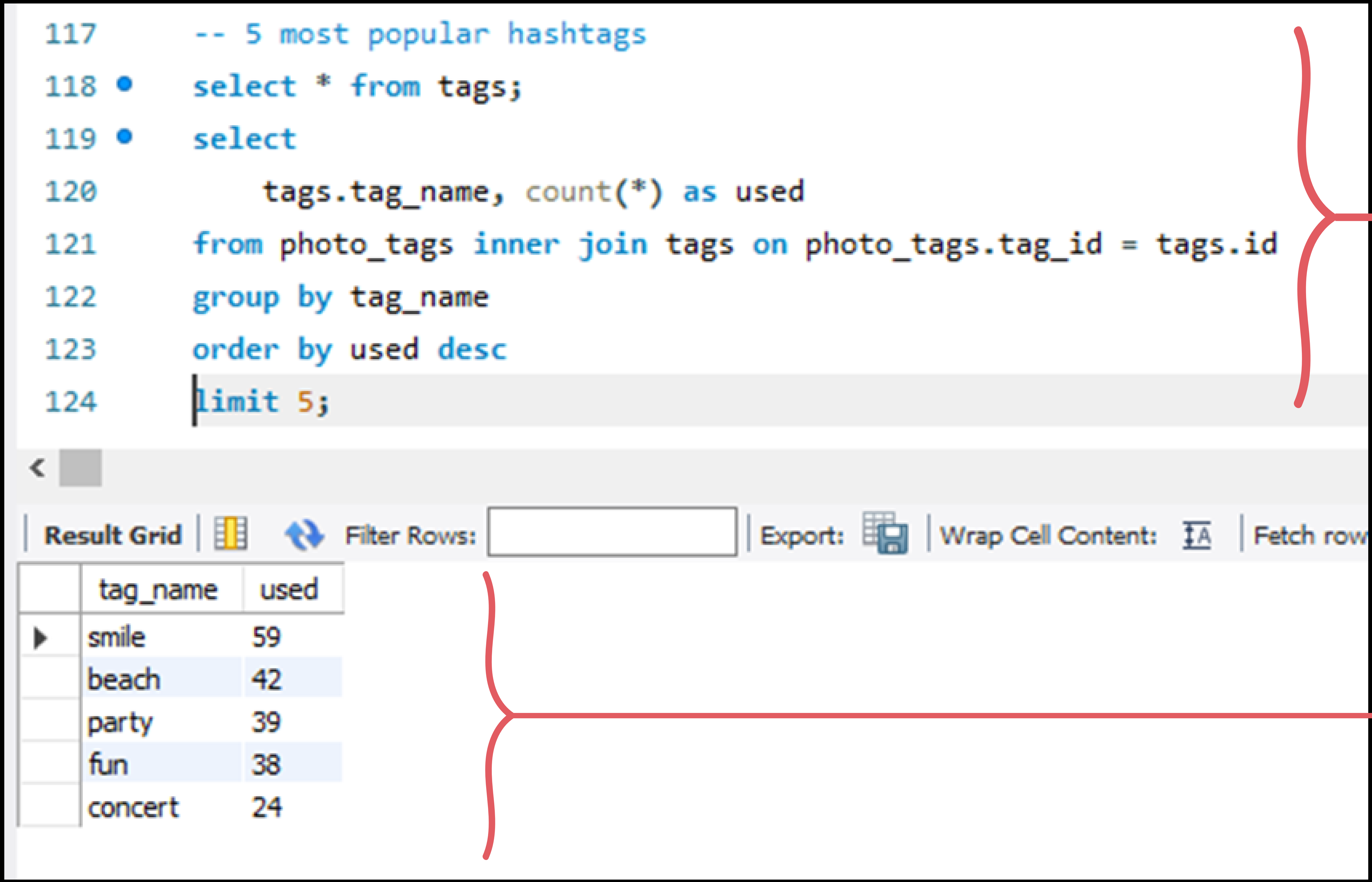
2.inner join tags table and photo_tags table on photo_tags.tag_id = tags.id

```
      ...inner join photo_tags on photo_tags.tag_id =
tags.id...
```

3.Group by tag_name, arrange entries in descending order based on `used` and finally select the first five entries using limit.

```
      ...group by tag_name order by used limit 5;
```





```sql
-- 5 most popular hashtags
select * from tags;
select
    tags.tag_name, count(*) as used
from photo_tags inner join tags on photo_tags.tag_id = tags.id
group by tag_name
order by used desc
limit 5;
```

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

Query

Top five hashtags

Ad Campaign Launch:

The team wants to know the best day of the week to launch ads.

Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

APPROACH:

1.Select dayname(created_at) as `day` and count(*) from `users` table.

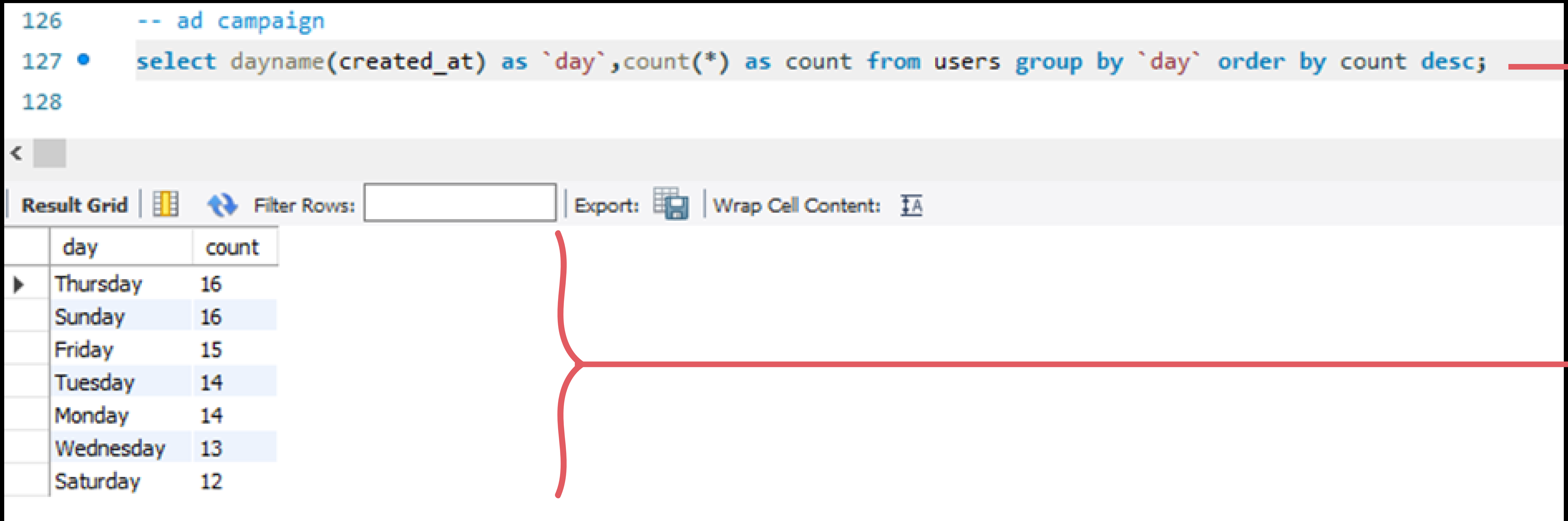
```
        select dayname(created_at) as `day`, count(*) as
count from users...
```

2.Group by `day`

```
        ...group by `day`...
```

3.Arrange the rows in descending order of the count so that the desired day can be retrieved

```
        ...order by count desc;
```



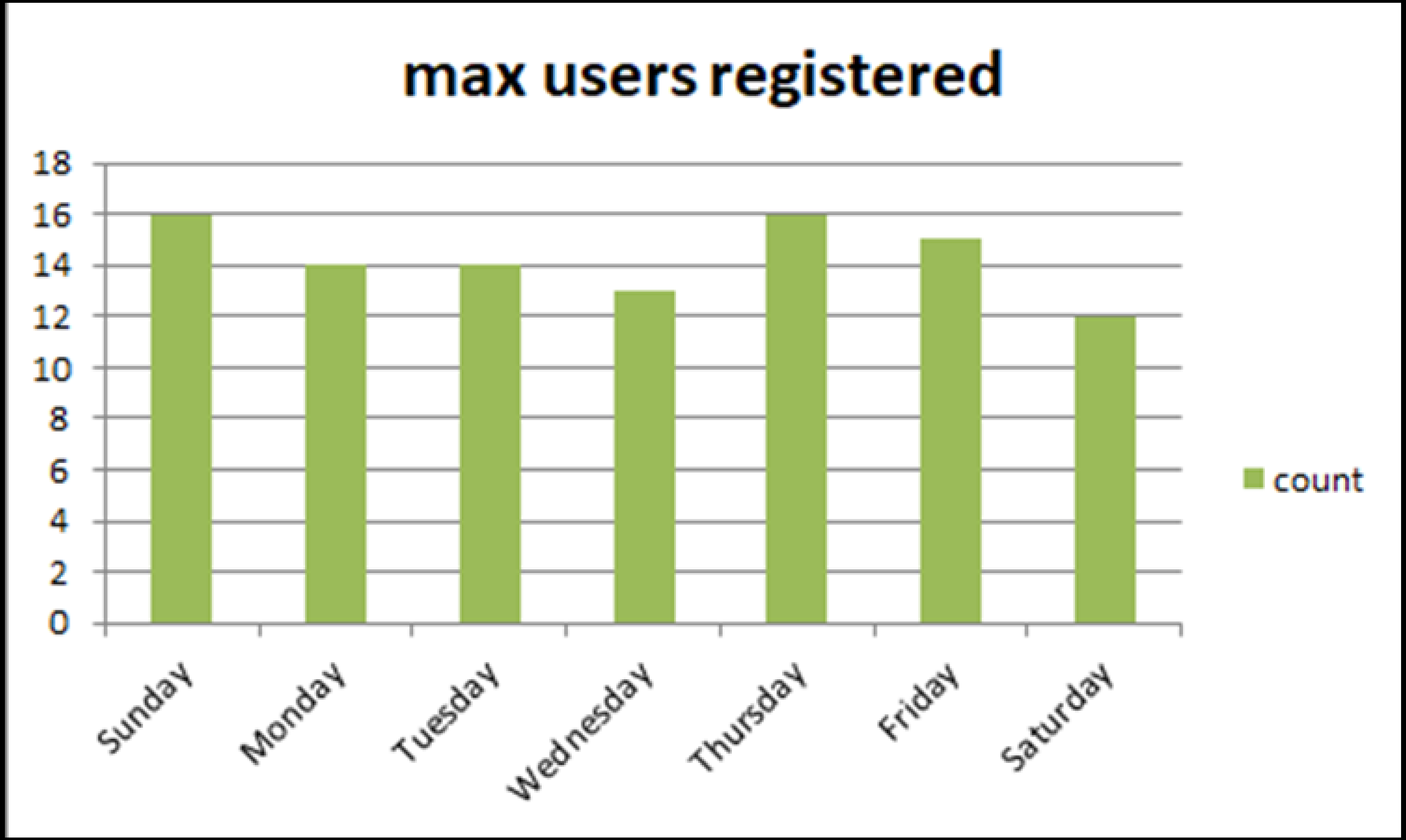
```
-- ad campaign
select dayname(created_at) as `day`,count(*) as count from users group by `day` order by count desc;
```

day	count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Query

No. of users that registered on a particular day

As seen above, Thursdays and Sundays are the best days to launch ads.



The above graph depicts the best days to launch ad campaign.

PART B

Investor Metrics

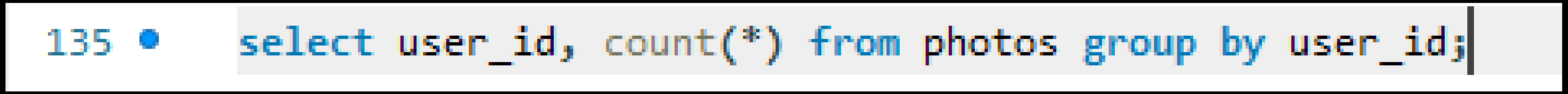
[software used: MySQL workbench 8.0 ce]

User Engagement:

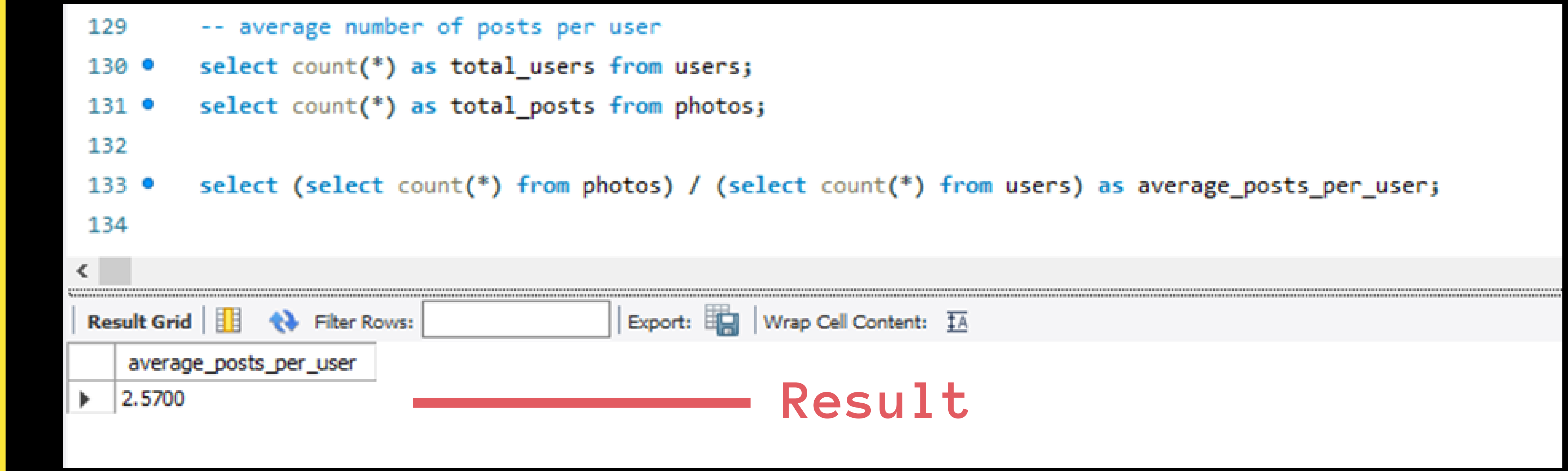
Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

Task: 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.

Query:



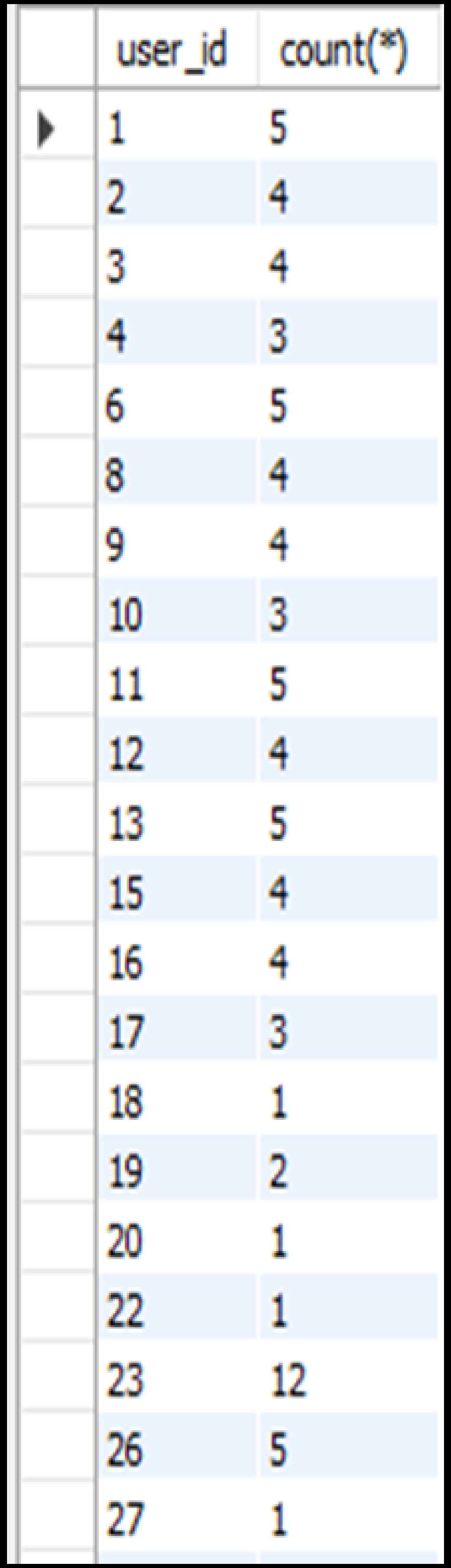
```sql
select user_id, count(*) from photos group by user_id;
```



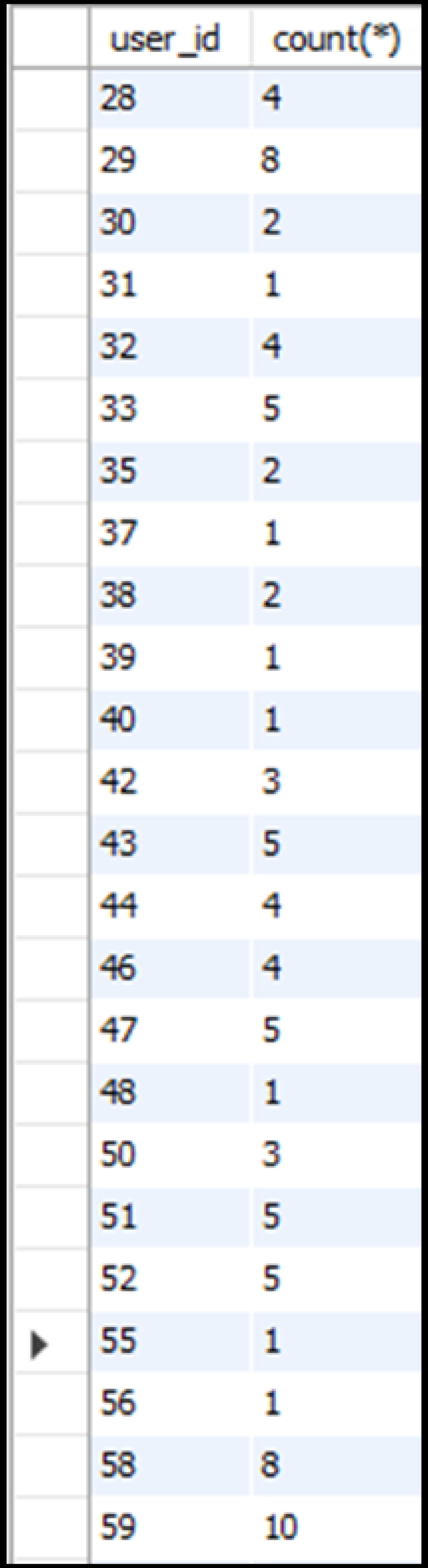
```sql
-- average number of posts per user
select count(*) as total_users from users;
select count(*) as total_posts from photos;

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

```
average_posts_per_user
2.5700
```

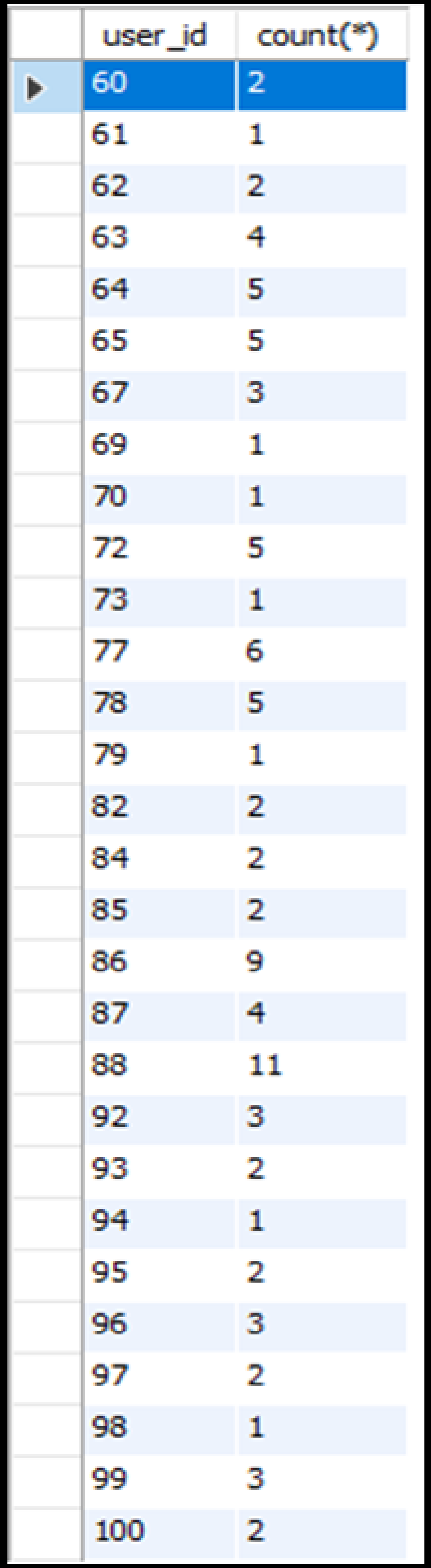
Result



user_id	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



user_id	count(*)
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



user_id	count(*)
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

The table shows each user and the number of posts created by them on the platform.

Bots & Fake Accounts:

Investors want to know if the platform is crowded with fake and dummy accounts.

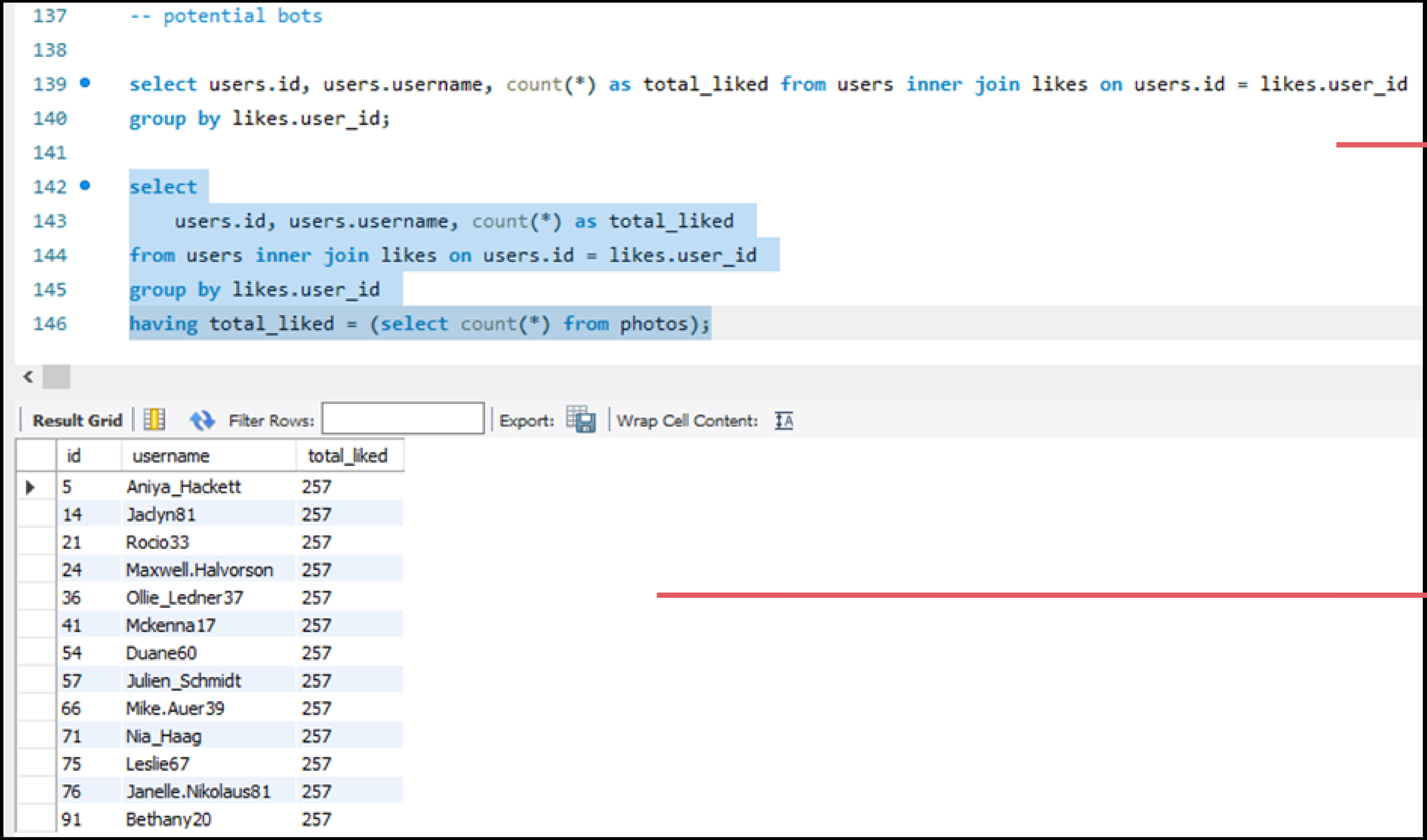
Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

APPROACH:

1.Select users.id, users.username, count(*) as total_liked from `users` table.

2.inner join users table and likes table on users.id = likes.user_id

3.Group by likes.user_id and using having keyword filter only those rows whose `total_liked` is same as that of total number of photos on instagram.



```sql
-- potential bots

select users.id, users.username, count(*) as total_liked from users inner join likes on users.id = likes.user_id
group by likes.user_id;

select
    users.id, users.username, count(*) as total_liked
from users inner join likes on users.id = likes.user_id
group by likes.user_id
having total_liked = (select count(*) from photos);
```

id	username	total_liked
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

Query

Potential Bots

Conclusion

THE TASK WAS PERFORMED USING BASIC AND ADVANCED TOPICS IN SQL. MYSQL WORKBENCH,A GRAPHICAL TOOL, WAS USED TO WRITE QUERIES AND OUTPUT THE DESIRED RESULTS.THE TASK HELPED TO PROVIDE DEEPER INSIGHTS FROM THE GIVEN DATASET.