Rajeev Kumar

INSTAGRAM

USER ANALYTICS

This project aims to analyze **Instagram user data** to provide valuable insights for marketing and investor decisions. The goal is to identify key trends, such as the most **loyal users**, **inactive users, contest winners**, **hashtag** popularity, and the best time for ad campaigns. Additionally, we analyze **user engagement** and detect potential bot activity to improve platform quality. The project is executed using **MySQL Workbench** to extract meaningful patterns from the data.

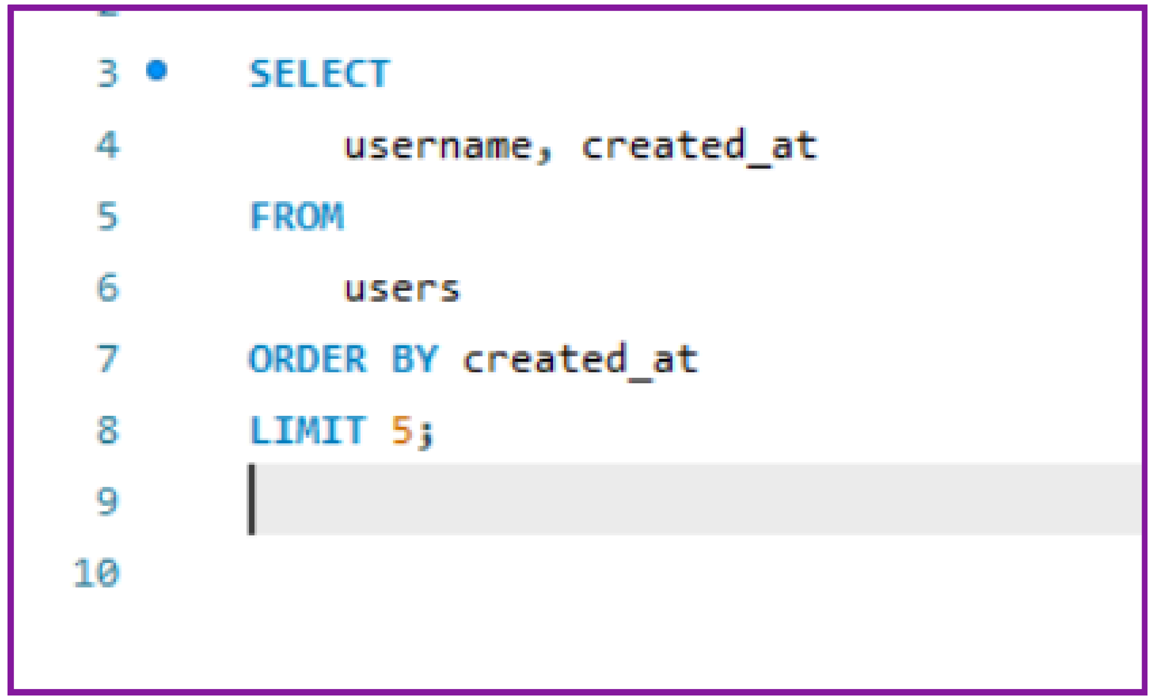
Project Description

This project focuses on analyzing Instagram user data to provide key insights for marketing and investor decisions. The goal is to extract meaningful information such as :

- Identifying the most loyal users.
- Finding inactive users and encouraging them to engage.
- Determining the contest winner based on likes.
- Researching the most popular hashtags.
- Analyzing the best day for launching ad campaigns.
- Measuring user engagement.
- Detecting potential bot accounts.

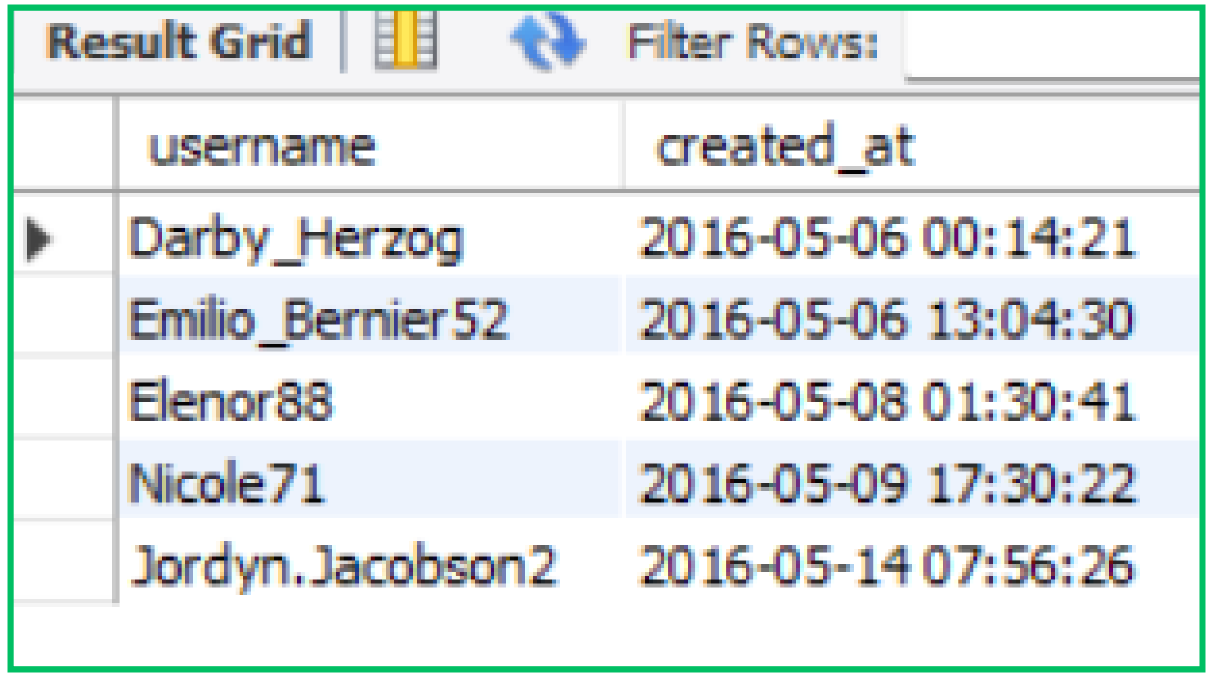
1. Identify the five oldest users on Instagram from the provided database.

SOLUTION



```
SELECT
    username, created_at
FROM
    users
ORDER BY created_at
LIMIT 5;
```

RESULT



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

1. **SELECT username, created_at**
   - I selected the username and created_at columns to see the users' names and when they joined.

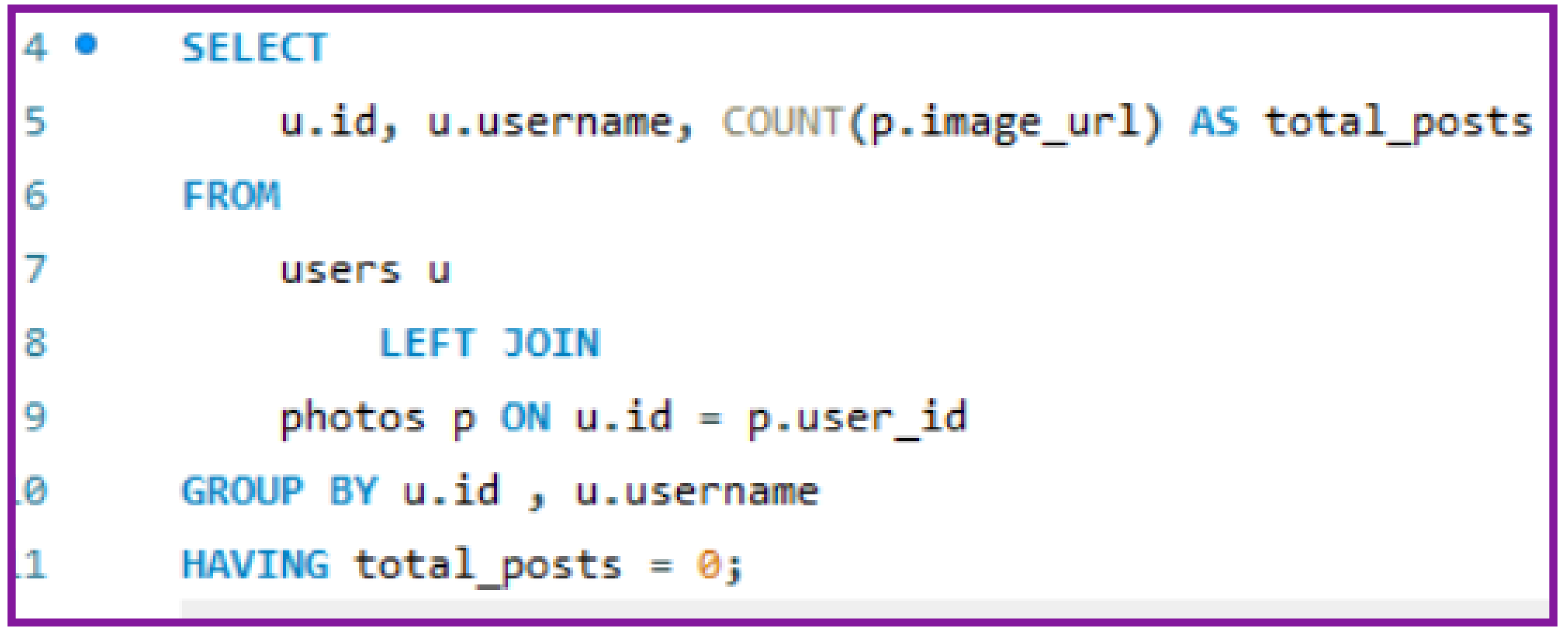
2. FROM users
- I used the users table because it has the required user data.

3. ORDER BY created_at
- I ordered the results by created_at to get the users who joined the earliest

4. LIMIT 5
- I limited the results to 5, so I get only the 5 oldest users

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

SOLUTION



```
SELECT
    u.id, u.username, COUNT(p.image_url) AS total_posts
FROM
    users u
        LEFT JOIN
    photos p ON u.id = p.user_id
GROUP BY u.id , u.username
HAVING total_posts = 0;
```

1. **SELECT u.id, u.username, COUNT(p.image_url) AS total_posts**
   - I selected the id and username from the users table, and also counted how many photos each user has by counting image_url from the photos table.

2. FROM users u

- I used the users table to get user details like id and username.

3. LEFT JOIN photos p ON u.id = p.user_id

- I joined the photos table with users to match each user with their photos using the user_id.

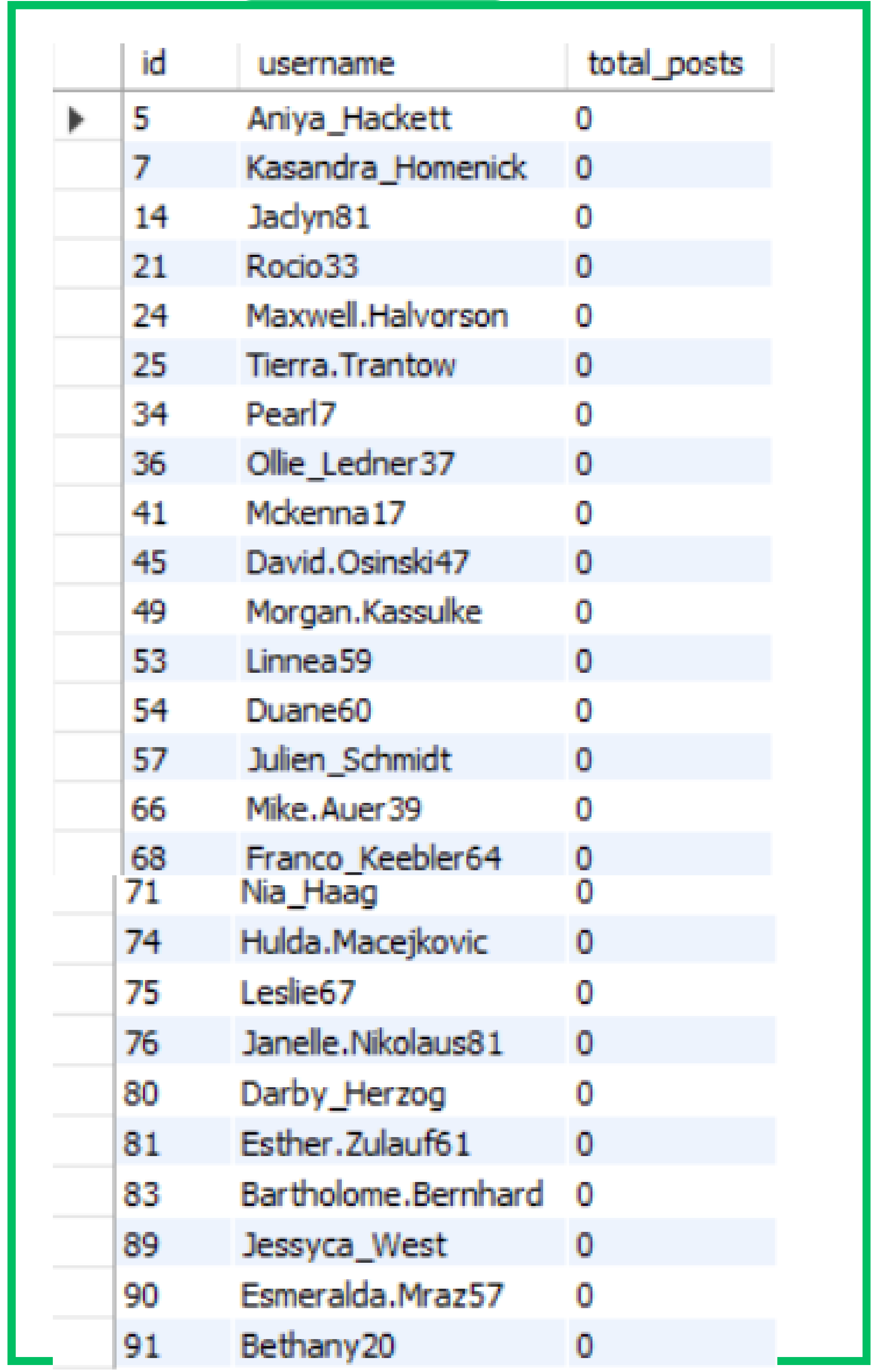
4. GROUP BY u.id, u.username

- I grouped the data by user id and username so I can calculate the total posts for each user.

5. HAVING total_posts = 0

- I used HAVING to filter the results and show only users with 0 posts.

RESULT



id	username	total_posts
5	Aniya_Hackett	0
7	Kasandra_Homenick	0
14	Jaclyn81	0
21	Rocio33	0
24	Maxwell.Halvorson	0
25	Tierra.Trantow	0
34	Pearl7	0
36	Ollie_Ledner37	0
41	Mckenna17	0
45	David.Osinski47	0
49	Morgan.Kassulke	0
53	Linnea59	0
54	Duane60	0
57	Julien_Schmidt	0
66	Mike.Auer39	0
68	Franco_Keebler64	0
71	Nia_Haag	0
74	Hulda.Macejkovic	0
75	Leslie67	0
76	Janelle.Nikolaus81	0
80	Darby_Herzog	0
81	Esther.Zulauf61	0
83	Bartholome.Bernhard	0
89	Jessyca_West	0
90	Esmeralda.Mraz57	0
91	Bethany20	0

3.

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

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

1. SELECT photo details and total likes:

- I selected photo_id, username, image_url, and counted how many users liked each photo.

2. JOIN likes, photos, and users

- I joined likes with photos to match likes with photos.
- Then, I joined photos with users to get the photo owner’s username.

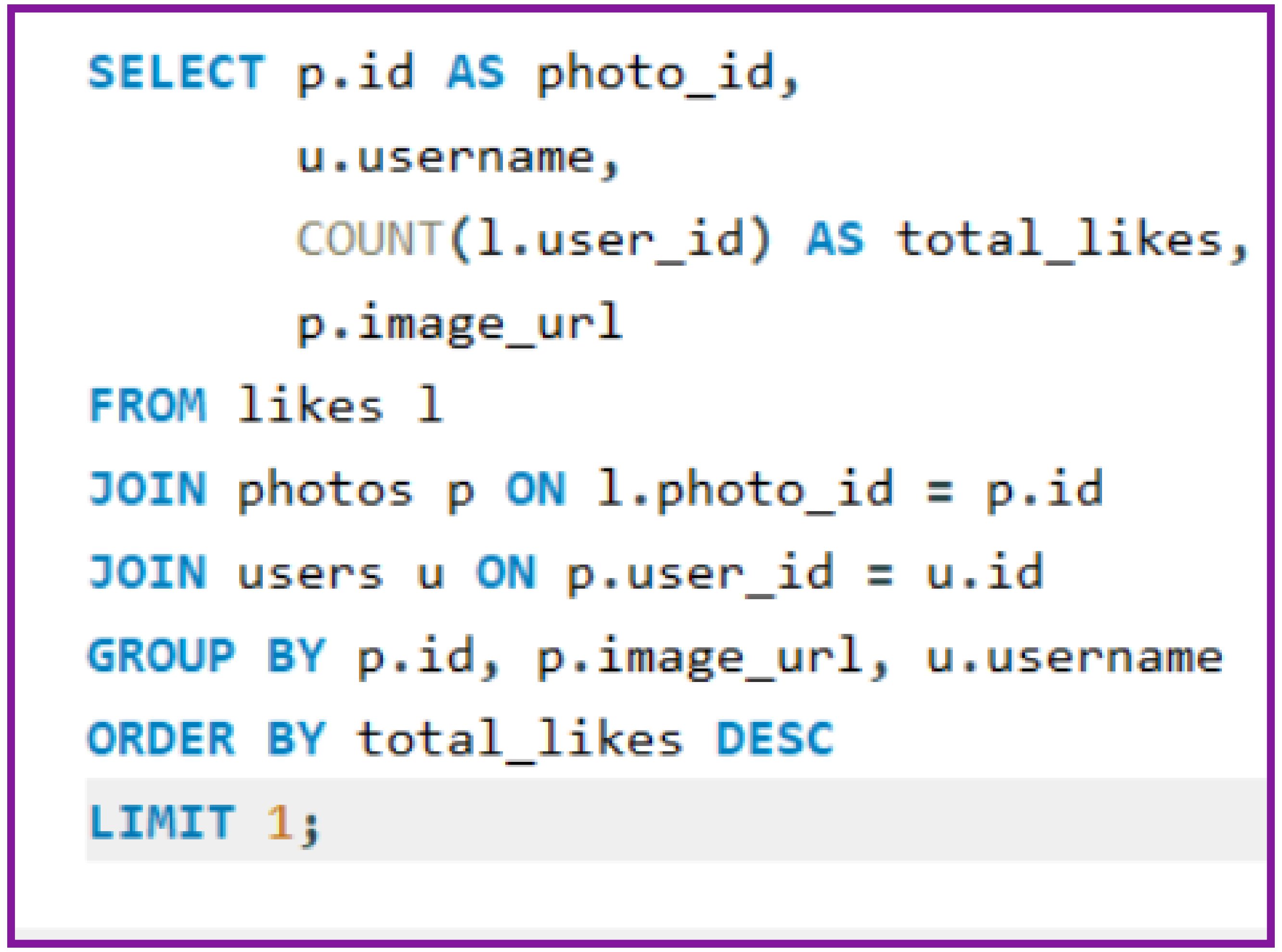
3. GROUP BY photo

- I grouped by photo_id, image_url, and username to count likes for each photo.

4. Sort and limit the result

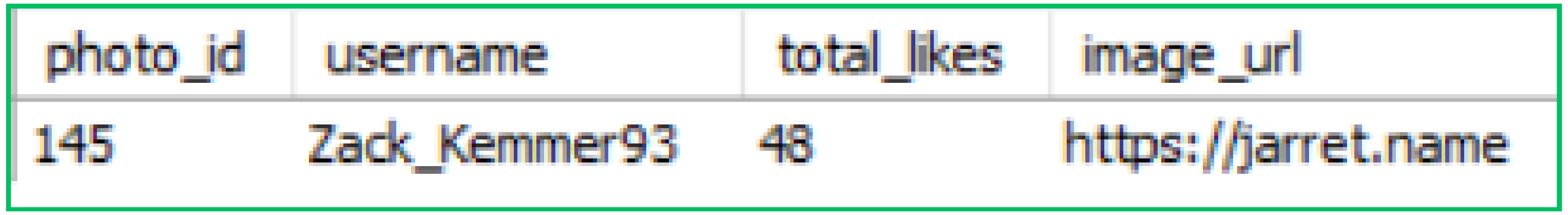
- I sorted by total_likes in descending order to find the most liked photo.
- I used LIMIT 1 to show only the top photo.

SOLUTION



```
SELECT p.id AS photo_id,
       u.username,
       COUNT(l.user_id) AS total_likes,
       p.image_url
FROM likes l
JOIN photos p ON l.photo_id = p.id
JOIN users u ON p.user_id = u.id
GROUP BY p.id, p.image_url, u.username
ORDER BY total_likes DESC
LIMIT 1;
```

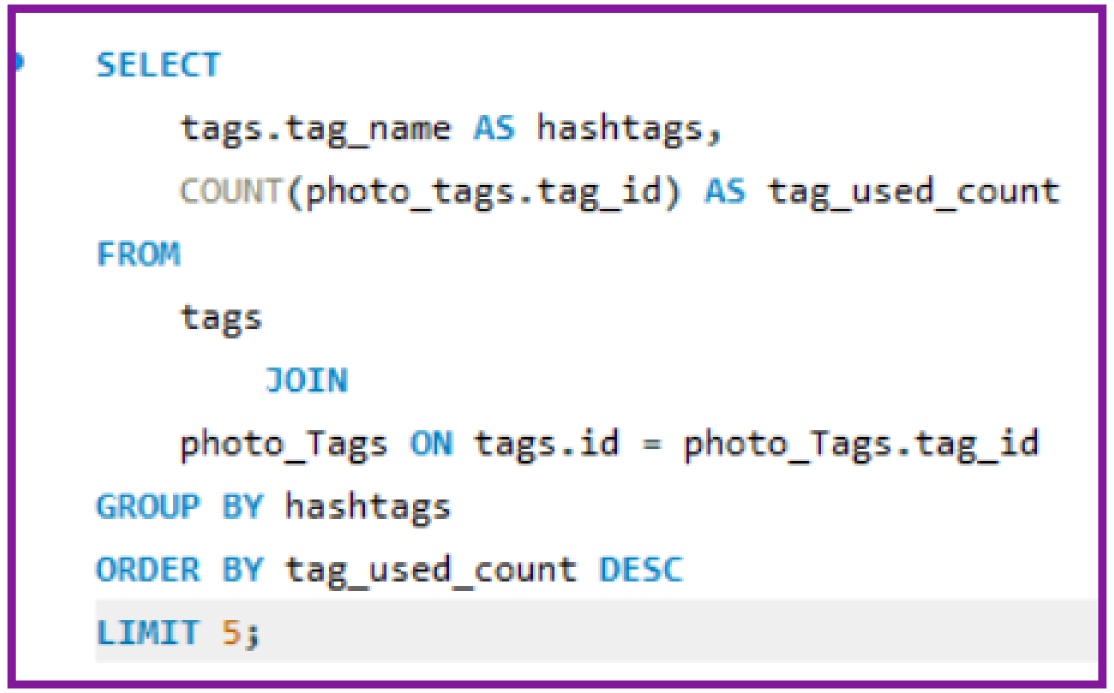
RESULT



photo_id	username	total_likes	image_url
145	Zack_Kemmer93	48	https://jarret.name

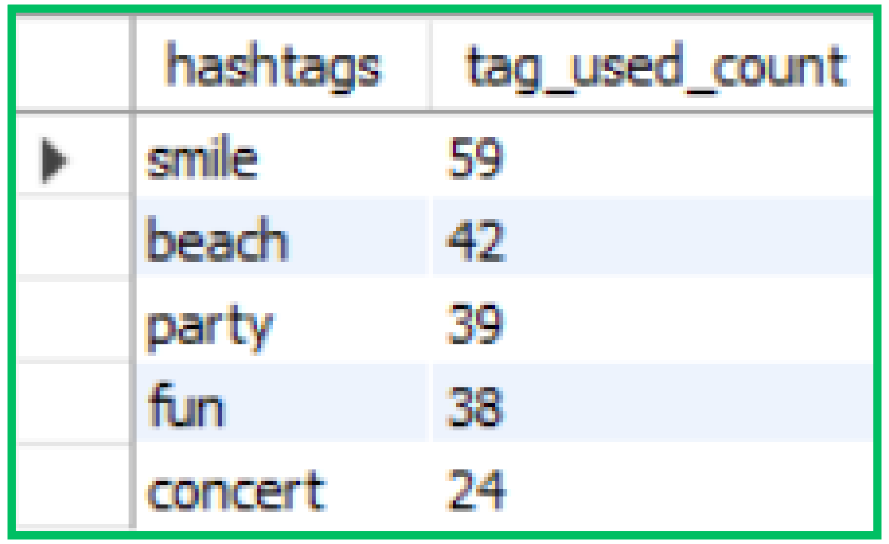
4. Identify and suggest the top five most commonly used hashtags on the platform.

SOLUTION



```
SELECT
    tags.tag_name AS hashtags,
    COUNT(photo_tags.tag_id) AS tag_used_count
FROM
    tags
        JOIN
    photo_Tags ON tags.id = photo_Tags.tag_id
GROUP BY hashtags
ORDER BY tag_used_count DESC
LIMIT 5;
```

RESULT



hashtags	tag_used_count
smile	59
beach	42
party	39
fun	38
concert	24

1. SELECT hashtags and count

- I selected the tag_name as hashtags and counted how many times each tag was used.

2. JOIN tags with photo_tags

- I joined tags with photo_tags to link hashtags to photos.

3. GROUP BY hashtags

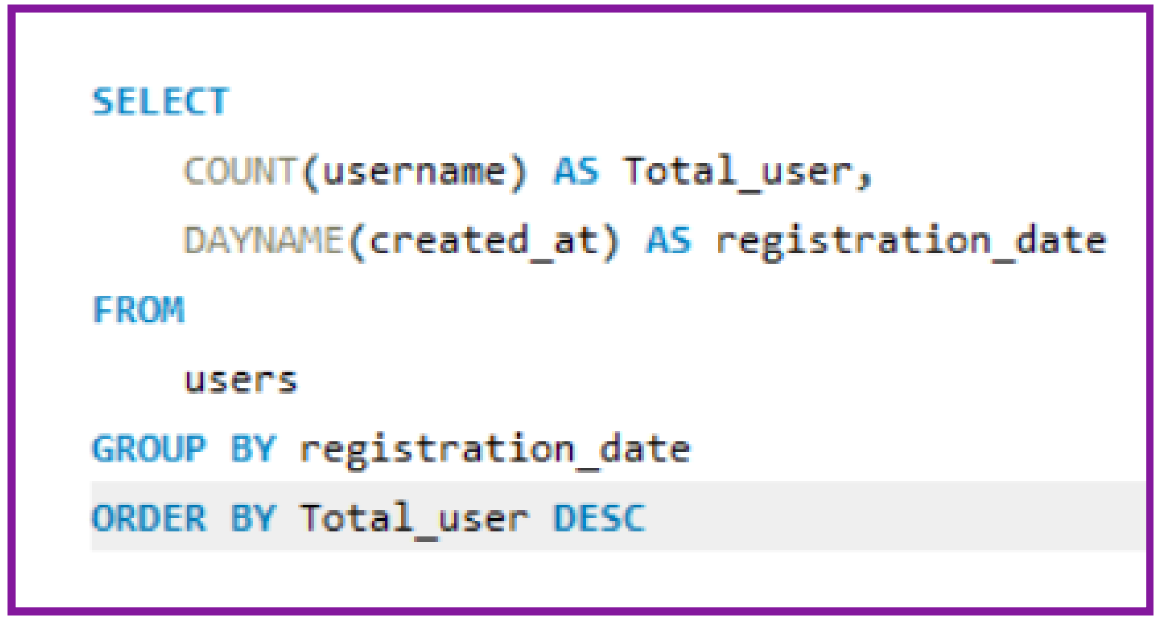
- I grouped by tag_name to count the usage of each hashtag.

4. ORDER BY tag usage, LIMIT 5:

- I sorted by the most used tags and limited the result to the top 5.

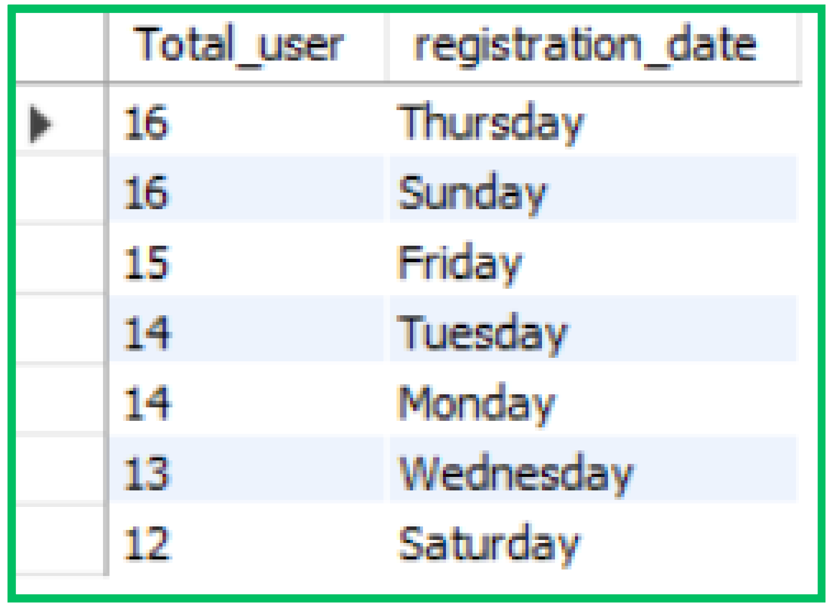
5. Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

SOLUTION



```
SELECT
    COUNT(username) AS Total_user,
    DAYNAME(created_at) AS registration_date
FROM
    users
GROUP BY registration_date
ORDER BY Total_user DESC
```

RESULT



Total_user	registration_date
16	Thursday
16	Sunday
15	Friday
14	Tuesday
14	Monday
13	Wednesday
12	Saturday

1. SELECT total users and registration day

- I counted the number of users and selected the day of the week they registered.

2. FROM users:

- I used the users table to get data about user registrations.

3. GROUP BY registration day

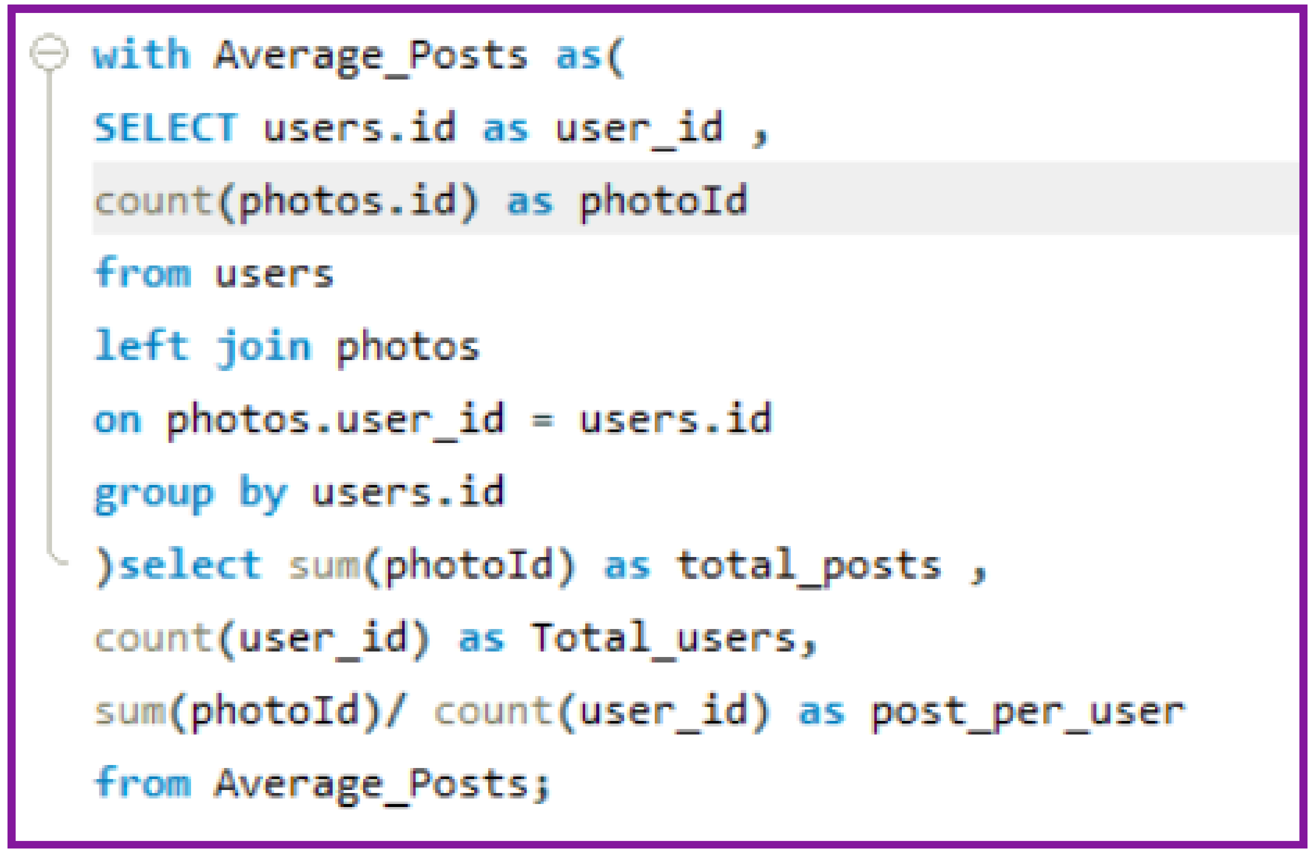
- I grouped by the day of the week users registered to see the count per day

4. ORDER BY total users

- I sorted by the number of users in descending order to find which day had the most registrations.

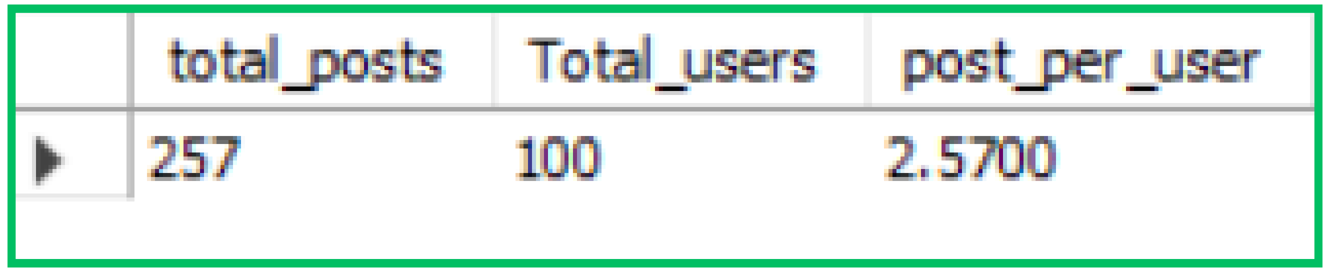
6. Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

SOLUTION



```
with Average_Posts as(
SELECT users.id as user_id ,
count(photos.id) as photoId
from users
left join photos
on photos.user_id = users.id
group by users.id
)select sum(photoId) as total_posts ,
count(user_id) as Total_users,
sum(photoId)/ count(user_id) as post_per_user
from Average_Posts;
```

RESULT



total_posts	Total_users	post_per_user
257	100	2.5700

1. WITH Average_posts

- I created a temporary table (Average_posts) to count the number of photos for each user using a LEFT JOIN between users and photos.

2. COUNT and GROUP BY

- I counted the photos for each user (COUNT(photos.id)) and grouped by user_id to get posts per user.

3. SELECT total posts, total users, posts per user

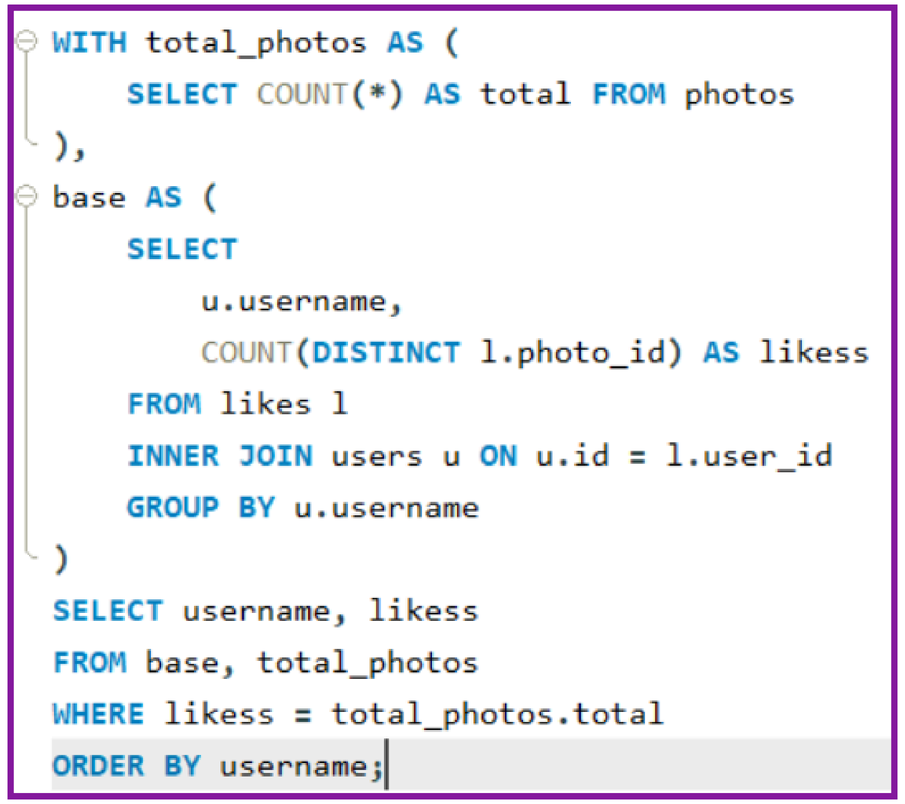
- I summed the total posts, counted the total users, and calculated the average posts per user by dividing total posts by total users

4. FROM Average_posts

- I used the Average_posts table to calculate the final results.

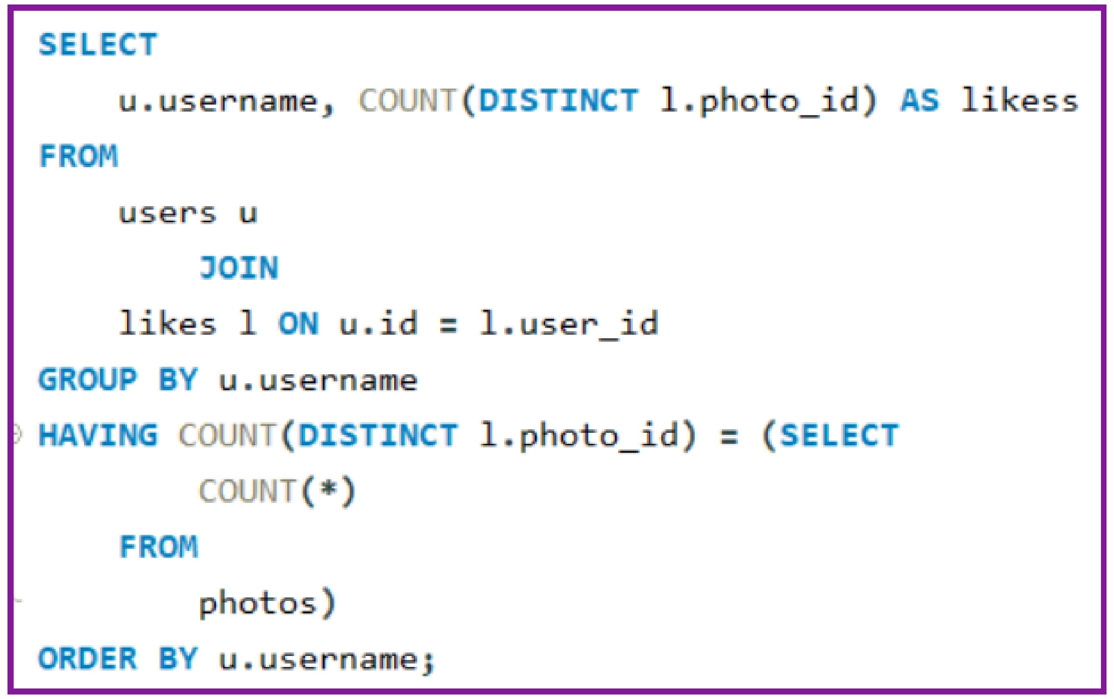
7. Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

SOLUTION **Way-1**



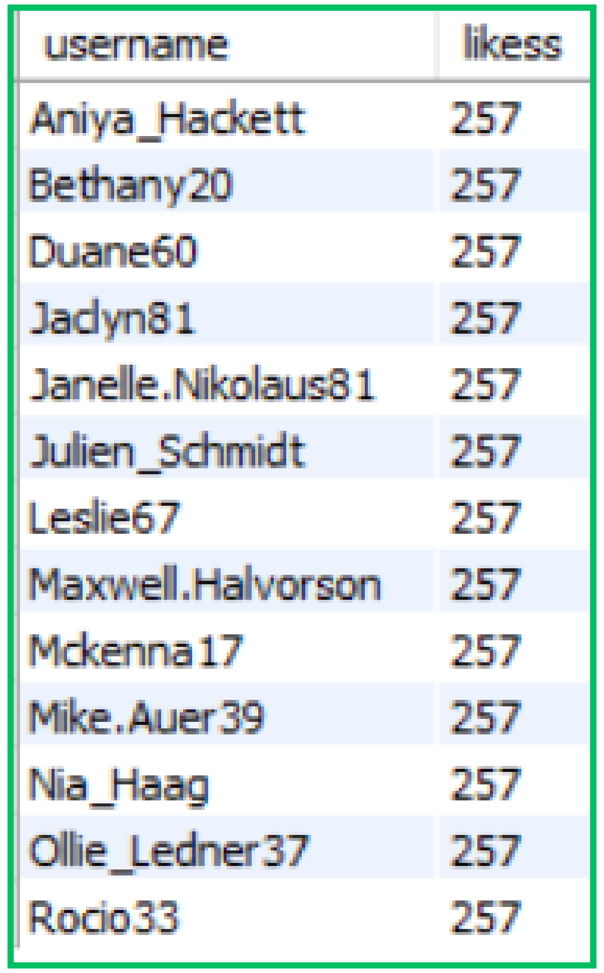
```
WITH total_photos AS (
    SELECT COUNT(*) AS total FROM photos
),
base AS (
    SELECT
        u.username,
        COUNT(DISTINCT l.photo_id) AS likess
    FROM likes l
    INNER JOIN users u ON u.id = l.user_id
    GROUP BY u.username
)
SELECT username, likess
FROM base, total_photos
WHERE likess = total_photos.total
ORDER BY username;
```

SOLUTION **Way-2**



```
SELECT
    u.username, COUNT(DISTINCT l.photo_id) AS likess
FROM
    users u
        JOIN
    likes l ON u.id = l.user_id
GROUP BY u.username
HAVING COUNT(DISTINCT l.photo_id) = (SELECT
        COUNT(*)
    FROM
        photos)
ORDER BY u.username;
```

RESULT



username	likess
Aniya_Hackett	257
Bethany20	257
Duane60	257
Jaclyn81	257
Janelle.Nikolaus81	257
Julien_Schmidt	257
Leslie67	257
Maxwell.Halvorson	257
Mckenna17	257
Mike.Auer39	257
Nia_Haag	257
Ollie_Ledner37	257
Rocio33	257

1. WITH total_photos

- I created a temporary table to count the total number of photos in the photos table.

2. WITH base

- I counted the distinct liked photos for each user by joining likes with users and grouping by username.

3. SELECT users who liked all photos

- I compared each user's total liked photos (likess) with the total number of photos.

4. Filter and sort results

- I used WHERE likess = total_photos.total to find users who liked every single photo and sorted them alphabetically.

Rajeev Kumar

THANK YOU

Through this project, I learned how to analyze Instagram user data using SQL. I successfully :

- Identified the oldest users and inactive users.
- Found the most liked photo and the top 5 hashtags.
- Discovered the day with the most registrations.
- Calculated the average posts per user.
- Detected potential bot users who liked all photos.

This analysis helped me understand user behavior on Instagram. It also improved my SQL skills, making me more confident in data analytics.