Instagram Clone SQL Project

By - Ujjwal Mishra

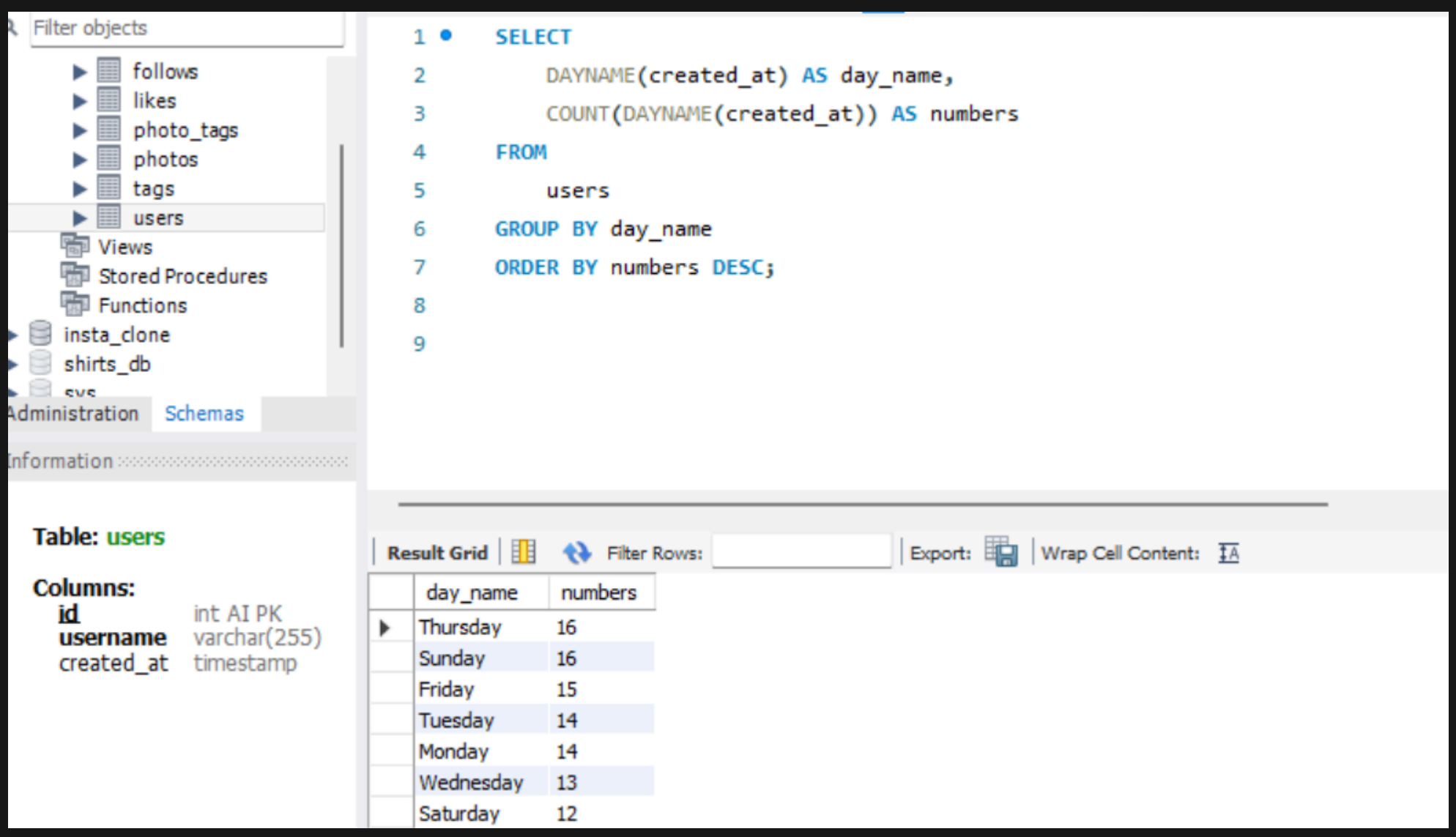
Objective :

The objective of this project is to leverage a cloned Instagram database to conduct a comprehensive SQL-based analysis, extracting actionable insights regarding user behaviour, content trends, and platform engagement.

This analysis will aim to identify key patterns and relationships within the data, providing a deeper understanding of users' behaviour on the Instagram platform.

1. We need to figure out when to schedule an ad campaign.

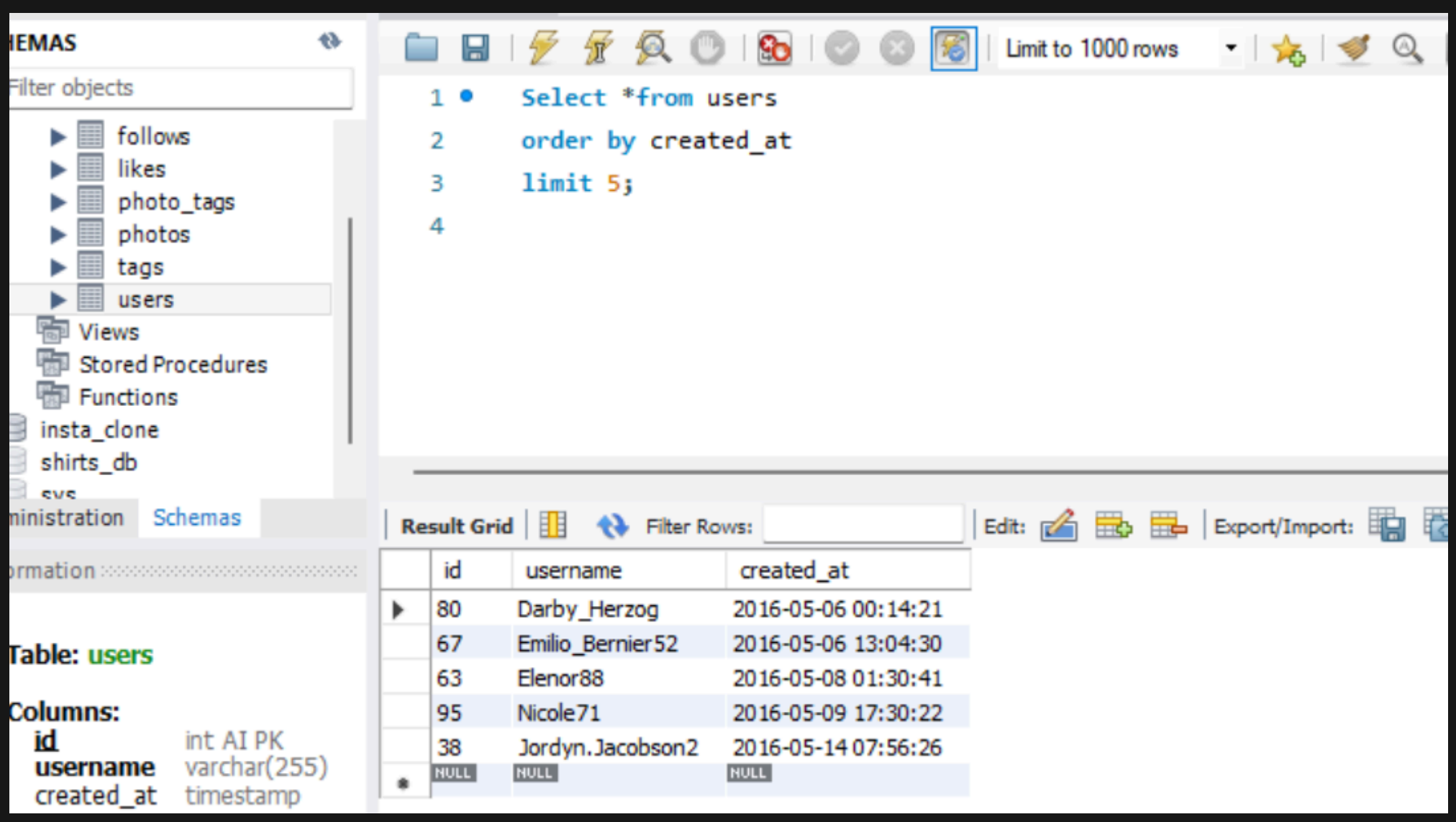
What day of the week do most users register on ?




```
Select DAYNAME (created_at) AS day_name,
COUNT (DAYNAME (created_at)) AS numbers
From users
GROUP BY day_name
ORDER BY numbers DESC;
```

2. We want to reward our users who have been around the longest.

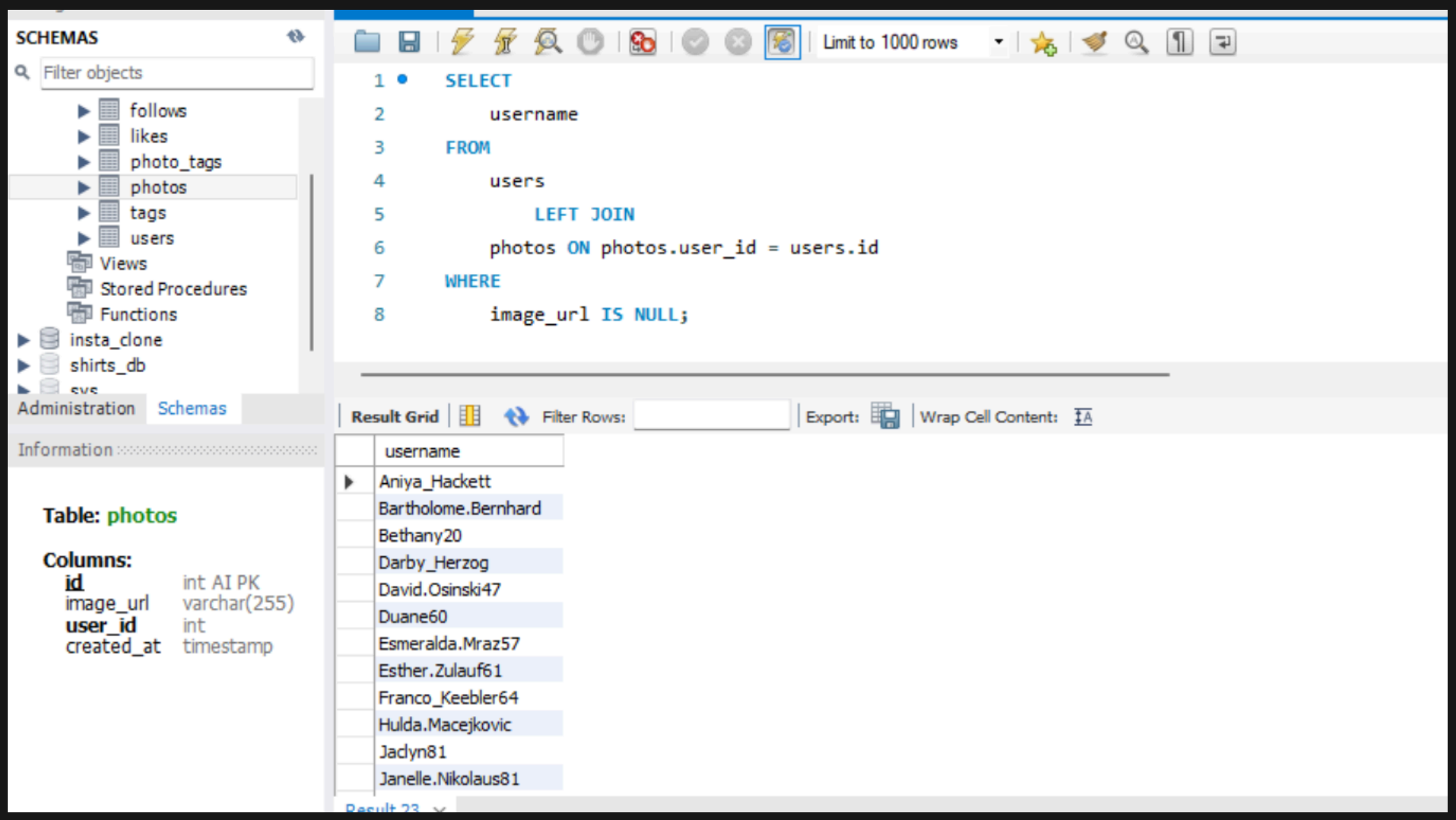
Find the 5 oldest users.




```
SELECT* FROM users
ORDER BY created_at
LIMIT 5;
```

3. We want to our target inactive users with an email campaign.

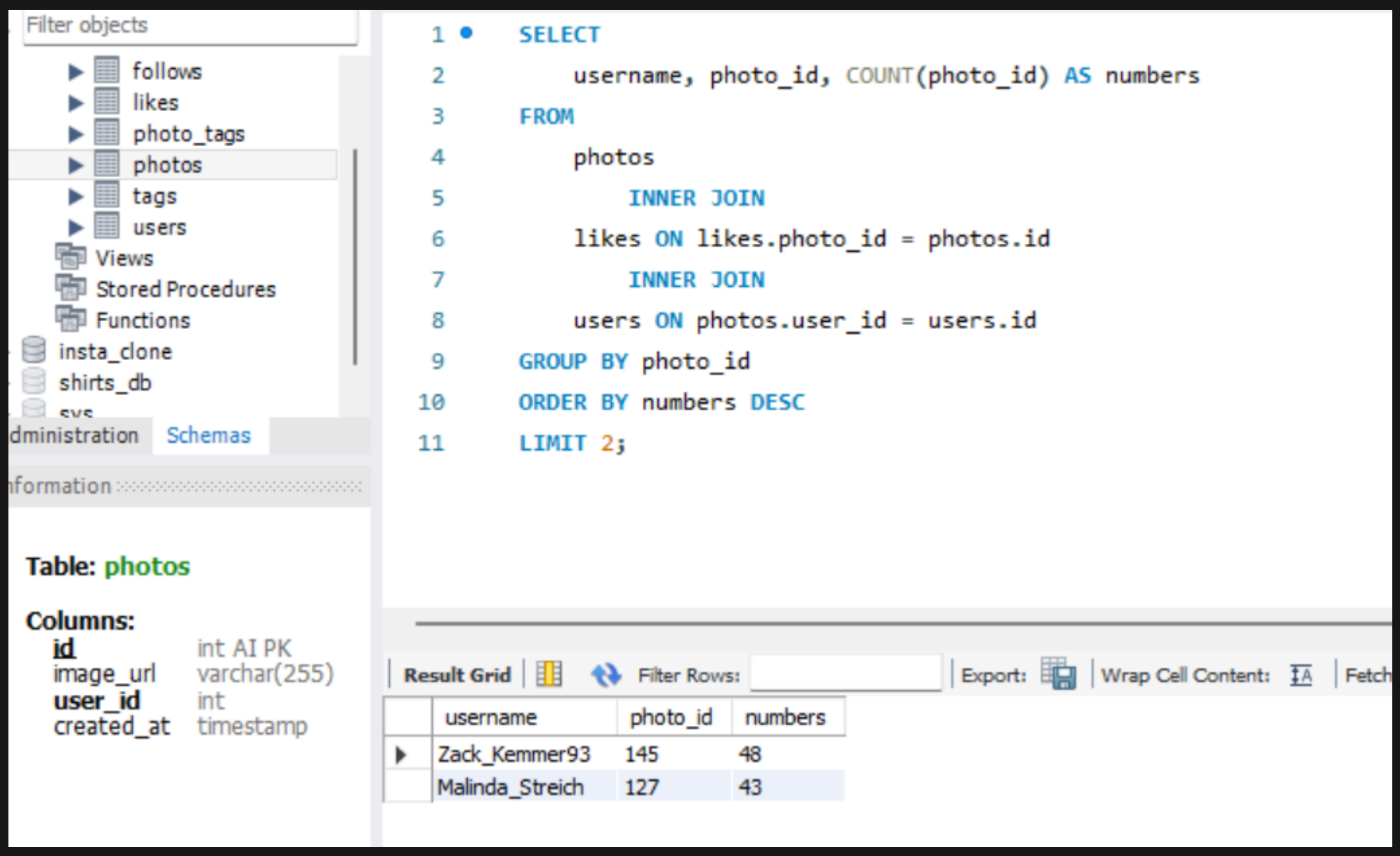
Find the users who have never posted a photo.




```
SELECT username FROM users
LEFT JOIN photos ON
photos.user_id = users.id
WHERE image_url IS NULL ;
```

4. We are running a new contest to see who can get the most the most like on a single photo.

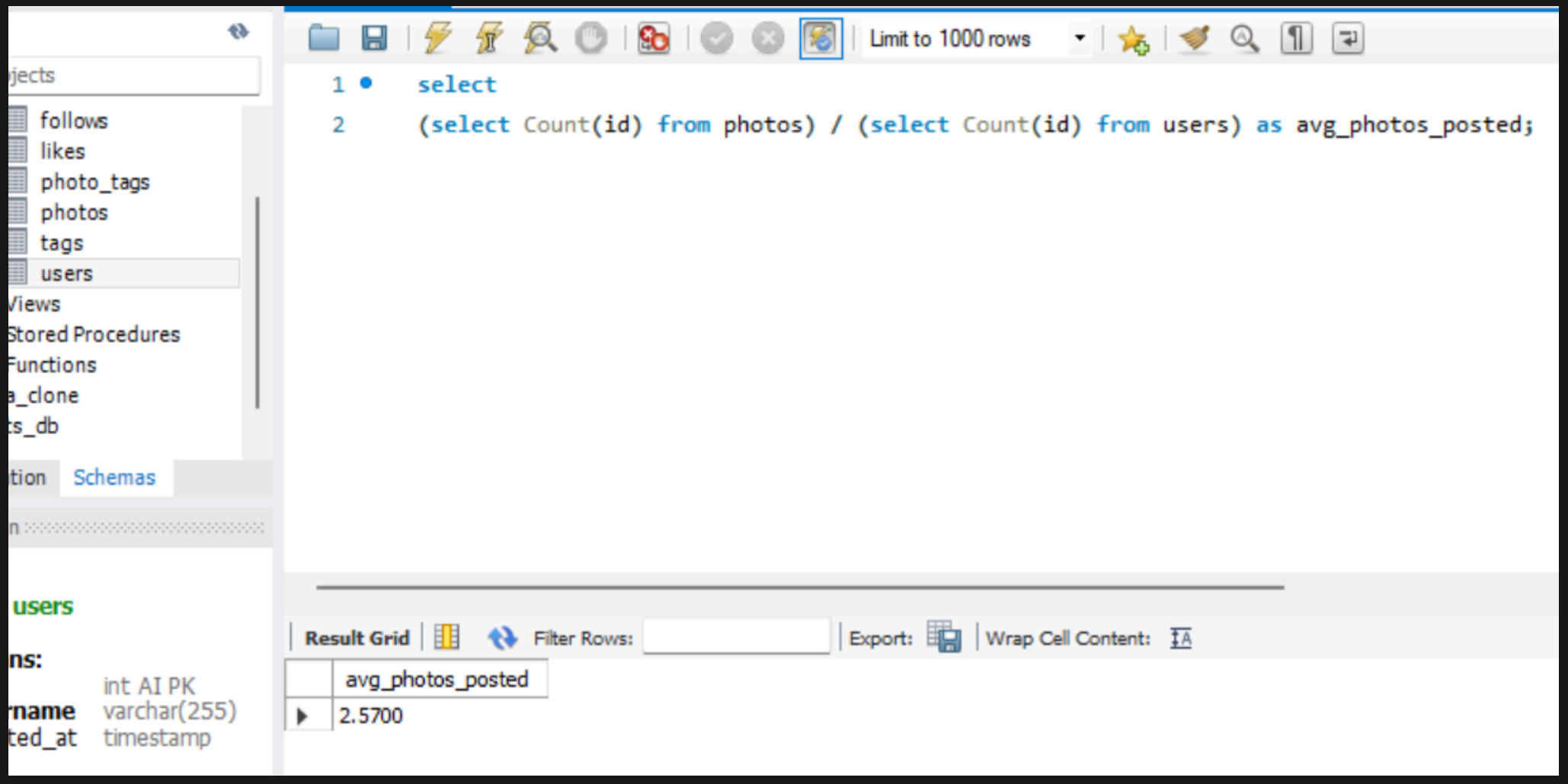
Who won ?




```
SELECT username, photo_id,
COUNT(photo_id) AS numbers FROM
photos
INNER JOIN likes ON likes.photo_id =
photos.id
INNER JOIN users ON photos,user_id =
photos.id
GROUP BY photo_id
ORDER BY numbers DESC
LIMIT 2 ;
```

5. Investors want to know.....

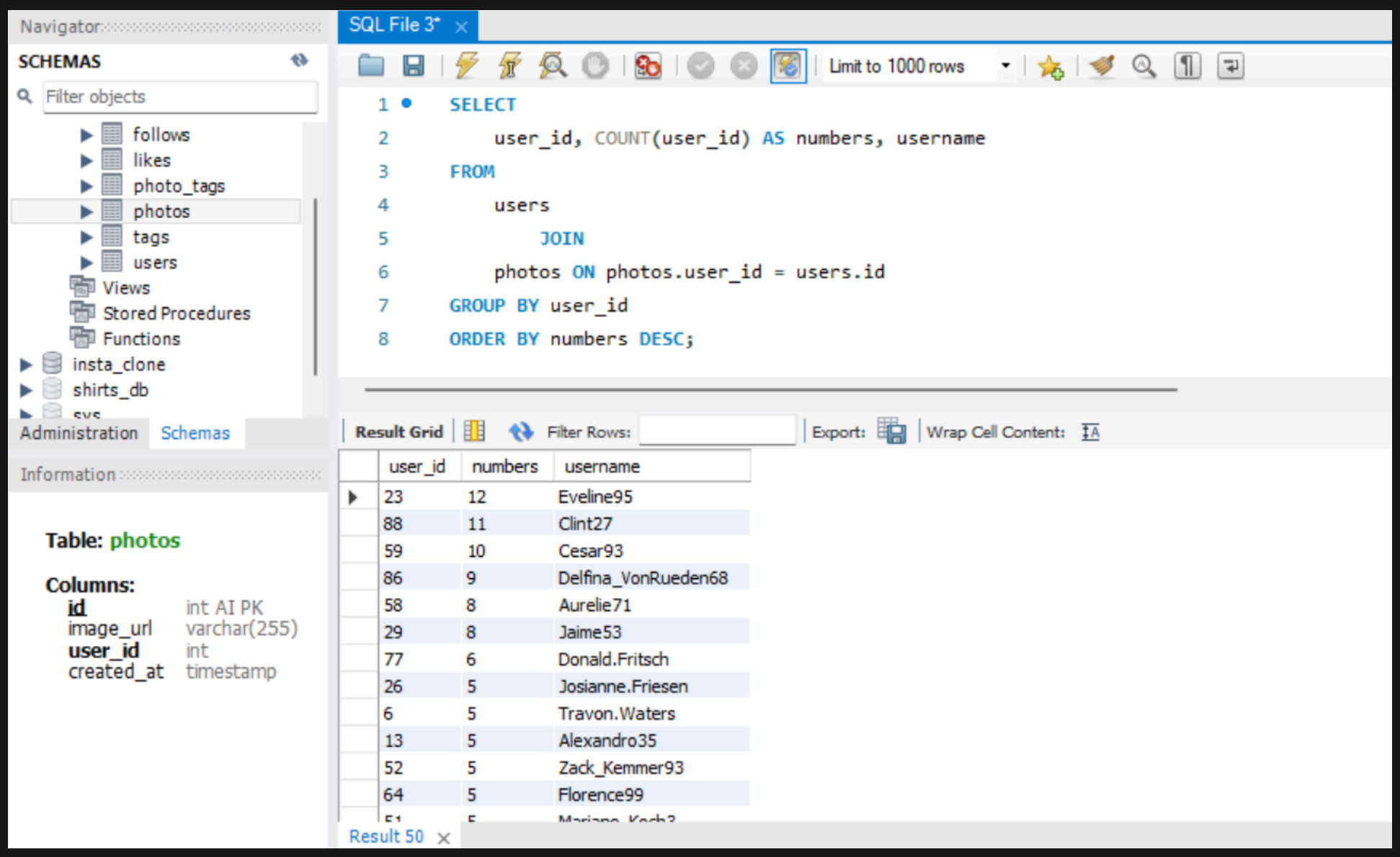
How many times does the average user posts ?




```
SELECT
(SELECT COUNT(id) FROM
photos / (SELECT COUNT (id)
FROM users)
AS avg_photos_posted ;
```

6. Investors want to know.....

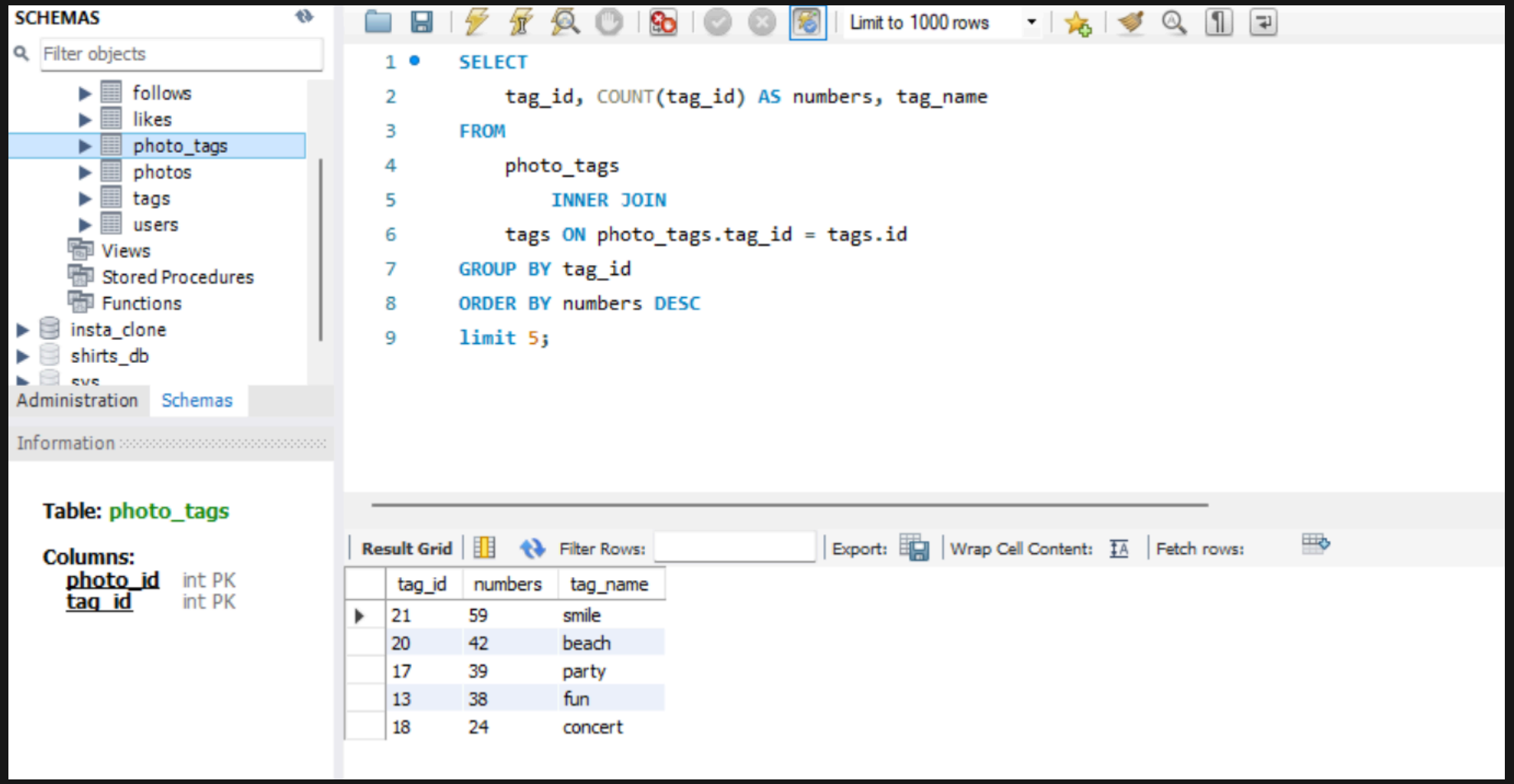
Numbers of photos posted by each user.




```
SELECT user_id, photo_id,
COUNT (user_id)
AS numbers, username
FROM
users
JOIN photos ON photos.user_id = users.id
GROUP BY user_id
ORDER BY numbers DESC ;
```

7. Brands want to know most popular hashtags.

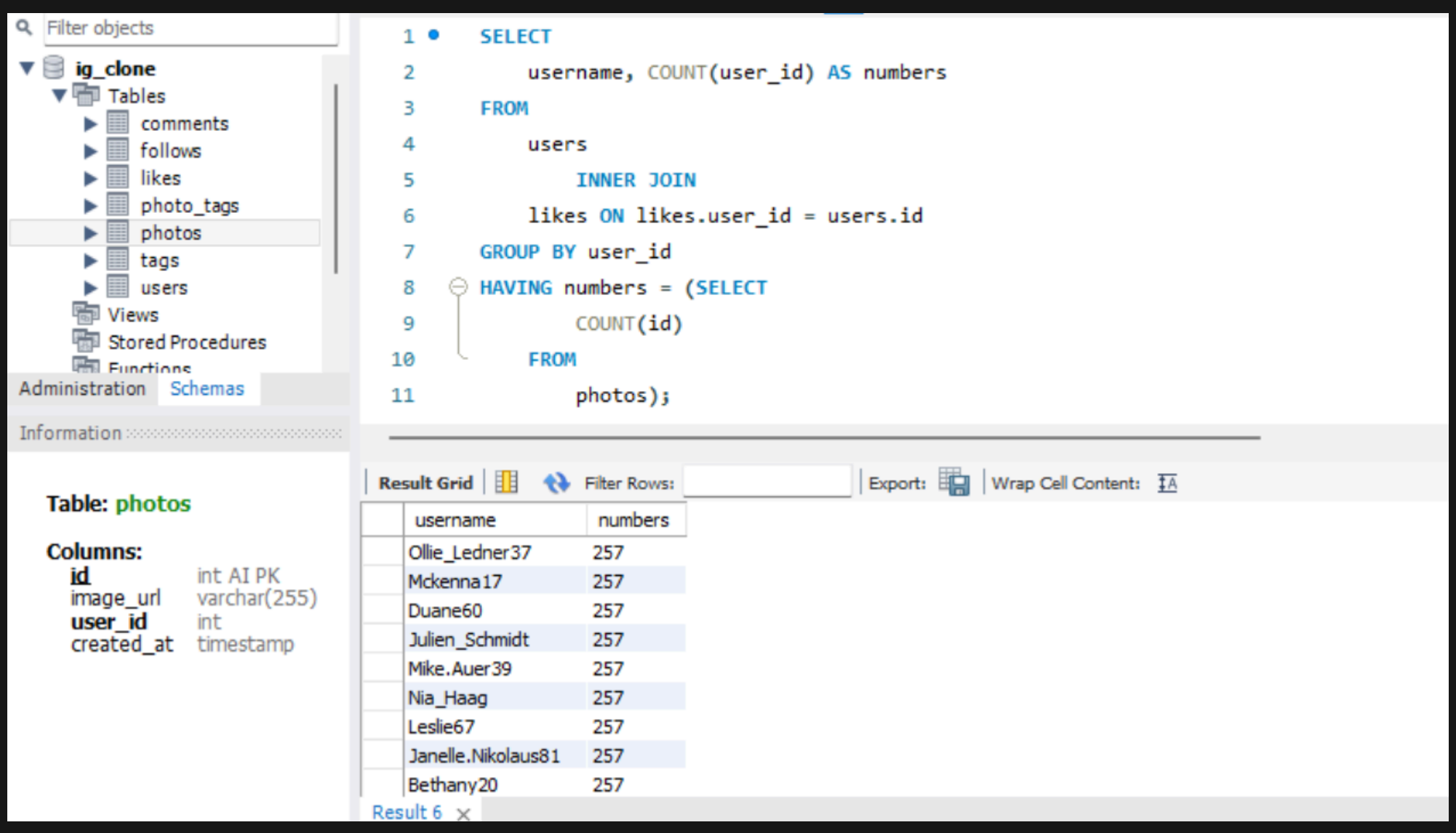
What are the top 5 most commonly used hashtags?




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

8. We have a small problem with bots on our site.

Find users who have likes every single photo on the site?



username	numbers
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bethany20	257

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

Thank You For Your Attention