Project: Instagram Clone Database Analysis

Author: Chetan Goudar

Introduction

- In this project, we will be analyzing the Instagram Clone database ('ig_clone') to extract valuable insights
- And answer specific questions related to user activity, content engagement, and more.
- Our goal is to provide meaningful data-driven recommendations to support decision-making for marketing, user engagement, and overall platform improvement.

'ig_clone' Database Schema

Key Tables

1. **Users Table:**
   - Fields: `id` (Primary Key), `username` (Unique), `created_at` (Timestamp).
   - Stores user information with a unique username and creation timestamp.
2. **Photos Table:**
   - Fields: `id` (Primary Key), `image_url`, `user_id` (Foreign Key), `created_at` (Timestamp).
   - Stores information about user-uploaded photos with image URL, linked to the Users table.
3. **Comments Table:**
   - Fields: `id` (Primary Key), `comment_text`, `photo_id` (Foreign Key), `user_id` (Foreign Key), `created_at` (Timestamp).
   - Stores user comments on photos, linked to both Users and Photos tables.
4. **Likes Table:**
   - Fields: `user_id` (Foreign Key), `photo_id` (Foreign Key), `created_at` (Timestamp).
   - Records likes on photos, with a composite primary key linking Users and Photos tables.
5. **Follows Table:**
   - Fields: `follower_id` (Foreign Key), `followee_id` (Foreign Key), `created_at` (Timestamp).
   - Represents user follow relationships, with a composite primary key linking Users table.
6. **Tags Table:**
   - Fields: `id` (Primary Key), `tag_name` (Unique), `created_at` (Timestamp).
   - Stores unique tags for categorizing photos.
7. **Photo_Tags Table:**
   - Fields: `photo_id` (Foreign Key), `tag_id` (Foreign Key).
   - Connects photos with associated tags, forming a composite primary key.

SQL Queries for Instagram Clone Database

Queries

1. Create an ER diagram or draw a schema for the given database.
2. Find the 5 oldest users.
3. Identify the day of the week most users register on.
4. Target inactive users in an ad campaign.
5. Determine the winner of a photo likes contest.
6. Calculate the average number of posts per user.
7. Find the top 5 most used hashtags.
8. Identify users who liked every photo on the site.
9. Discover users who have never commented on a photo.
10. Find users who have never commented on any photo or have commented on every photo.
11. Demonstrate the top 30 usernames to the company who have posted photos in the range of 3 to 5.
12. Can you help me find the users whose name starts with c and ends with any number and have posted the photos as well as liked the photos?
13. Find the users who have created Instagram id in May and select top 5 newest Joines from it?

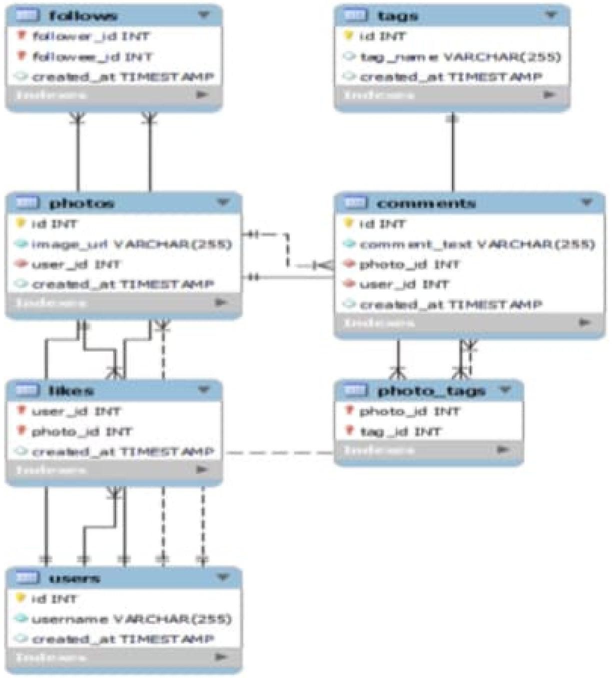
Q.1) ER Diagram for ig_clone Database

Entities:

- Users: Registered users of the app with profile information.
- Photos: Images uploaded by users with associated metadata.
- Tags: Keywords assigned to photos for categorization and organization.
- Follows: User-to-user relationships indicating who follows whom.
- Likes: User-to-photo relationships indicating user preferences.

Relationships:

- One-to-One Relationship: Each user has one profile photo, and each photo belongs to one user.
- One-to-Many Relationship: A user uploads multiple photos, but each photo belongs to only one user.
- Many-to-Many Relationship: A photo has multiple tags, and a tag can be applied to multiple photos.
- One-to-Many Relationship: A user follows multiple users, but each user can only be followed by other users.
- Many-to-Many Relationship: A user likes multiple photos, and a photo can be liked by multiple users.



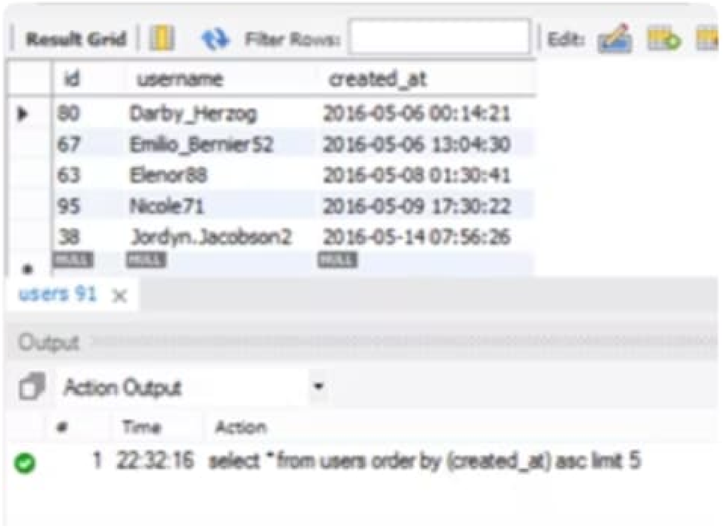
Q.2) We want to reward the user who has been around the longest, Find the 5 oldest users.

Code:

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

Insights:

Recognizing and rewarding long-time users is crucial for user retention. These users have been a part of the platform since its early days, showcasing loyalty and engagement.



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

```
#  Time      Action
1  22:32:16  select * from users order by (created_at) asc limit 5
```

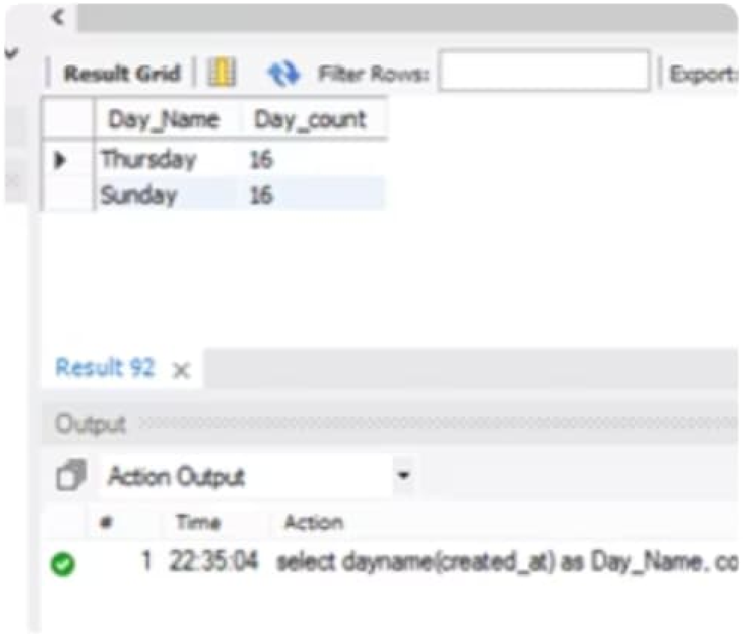
Q.3) To understand when to run the ad campaign, figure out the day of the week most users register on?

Code:

```
SELECT DAYNAME(created_at) AS Day_Name,
COUNT(*) AS Day_count

FROM users

GROUP BY Day_Name

LIMIT 2;
```

Insights:

Knowing the days of the week when most users register is valuable for planning targeted ad campaigns. This data can help optimize ad placement and timing to reach a larger audience.



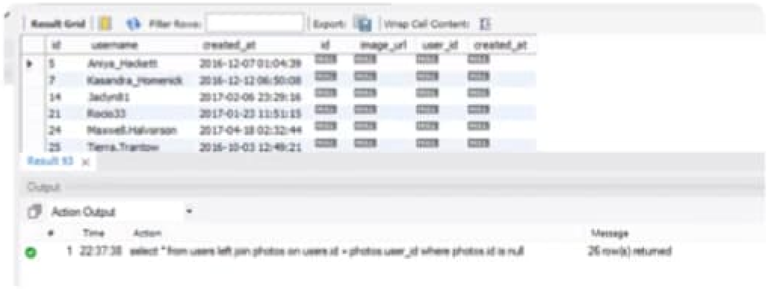
Day_Name	Day_count
Thursday	16
Sunday	16

```
#   Time       Action
1   22:35:04   select dayname(created_at) as Day_Name, co
```

Q.4) To target inactive users in an email ad campaign, find the users who have never posted a photo.

Code:

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



```
Result Grid | Filter Rows: | Export: | Wrap Cell Content:
   id  username            created_at           id    image_url  user_id  created_at
▶  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
Result 93 ×

Output
Action Output
   #  Time      Action                                                                              Message
●  1  22:37:38  select * from users left join photos on users.id = photos.user_id where photos.id is null   26 row(s) returned
```

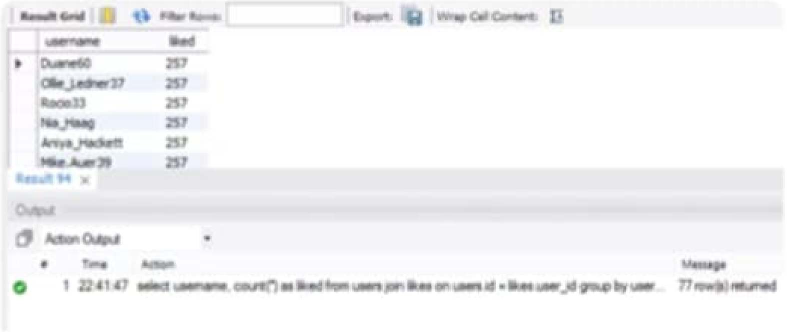
Insights:

Identifying users who have never posted a photo allows for targeted email ad campaigns to re-engage and encourage them to contribute content, enhancing overall platform activity.

Q.5) Suppose you are running a contest to find out who got the most likes on a photo. Find out who won?

Code:

```
SELECT username, COUNT(*) AS liked
FROM users
JOIN likes ON users.id = likes.user_id
GROUP BY user_id
ORDER BY liked DESC;
```



username	liked
Duane60	257
Ollie_Ledner37	257
Rocio33	257
Nia_Haag	257
Aniya_Hackett	257
Mike.Auer39	257

```
#  Time      Action                                                                              Message
1  22:41:47  select username, count(*) as liked from users join likes on users.id = likes.user_id group by user...  77 row(s) returned
```

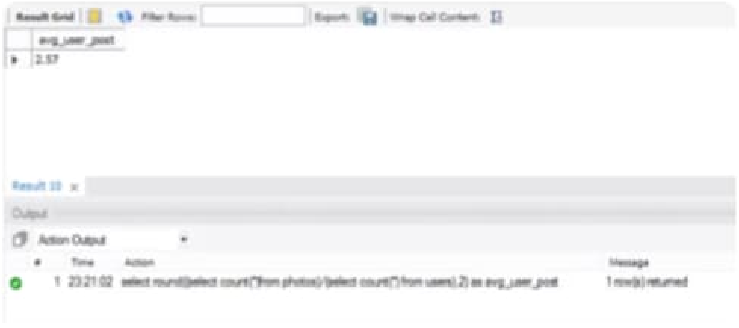
Insights:

Running contests based on likes fosters engagement. Recognizing users with the most likes not only boosts their profile but also encourages healthy competition and interaction on the platform.

Q.6) The investors want to know how many times does the average user post.

Code:

```
SELECT ROUND((SELECT COUNT(*) FROM photos) /
(SELECT COUNT(*) FROM users), 2) AS
avg_user_post;
```



```
avg_user_post
2.57
```

```
#  Time      Action                                                                          Message
1  23:21:02  select round((select count(*)from photos)/(select count(*) from users),2) as avg_user_post  1 row(s) returned
```

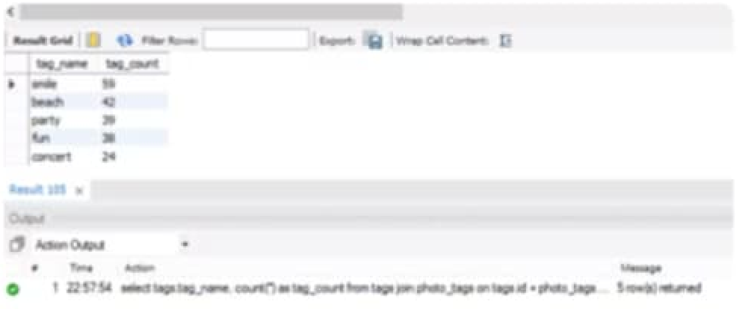
Insights:

Investors seeking information on user activity find the average post frequency valuable. This metric indicates how actively users are contributing content, impacting the overall vibrancy of the platform.

Q.7) A brand wants to know which hashtag to use on a post and find the top 5 most used hashtags.

Code:

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



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

```
#   Time      Action                                                                                     Message
1   22:57:54  select tags.tag_name, count(*) as tag_count from tags join photo_tags on tags.id = photo_tags...   5 row(s) returned
```

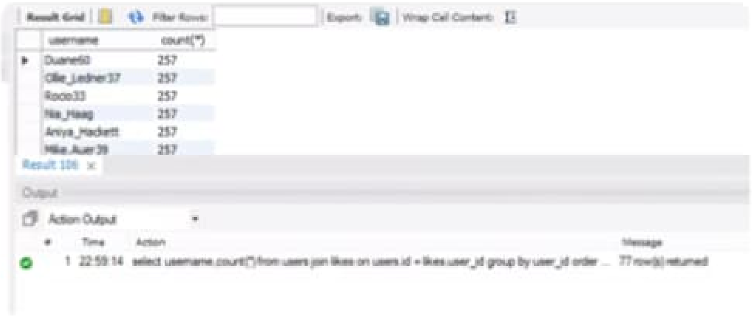
Insights:

For brand strategy, understanding the most used hashtags provides insights into popular trends and user interests. Brands can leverage these hashtags to maximize the visibility of their content.

Q.8) To find out if there are bots, find users who have liked every single photo on the site.

Code:

```
SELECT username, COUNT(*)
FROM users
JOIN likes ON users.id = likes.user_id
GROUP BY user_id
ORDER BY COUNT(*) DESC;
```



username	count(*)
Duane60	257
Ollie_Ledner37	257
Rocio33	257
Nia_Haag	257
Aniya_Hackett	257
Mike.Auer39	257

```
#   Time      Action                                                                                     Message
1   22:59:14  select username,count(*) from users join likes on users.id = likes.user_id group by user_id order ...   77 row(s) returned
```

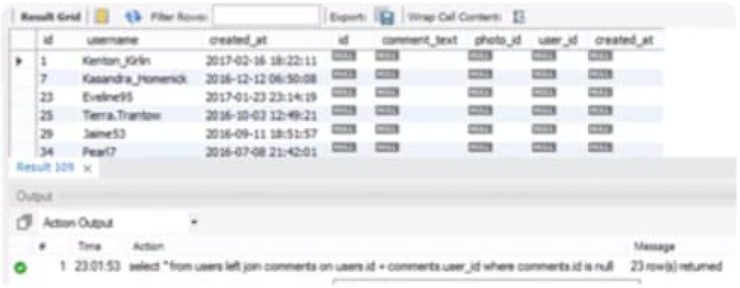
Insights:

Identifying users who have liked every single photo raises awareness about potential bot activity. This information is crucial for maintaining the integrity and authenticity of user engagement on the platform.

Q.9) To know who the celebrities are, find users who have never commented on a photo.

Code:

```
SELECT *
FROM users
LEFT JOIN comments ON users.id = comments.user_id
WHERE comments.id IS NULL;
```



id	username	created_at	id	comment_text	photo_id	user_id	created_at
1	Kenton_Kirlin	2017-02-16 18:22:11	NULL	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL	NULL
23	Eveline95	2017-01-23 23:14:19	NULL	NULL	NULL	NULL	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL	NULL
29	Jaime53	2016-09-11 18:51:57	NULL	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                                                            Message
1  23:01:53  select * from users left join comments on users.id = comments.user_id where comments.id is null   23 row(s) returned
```

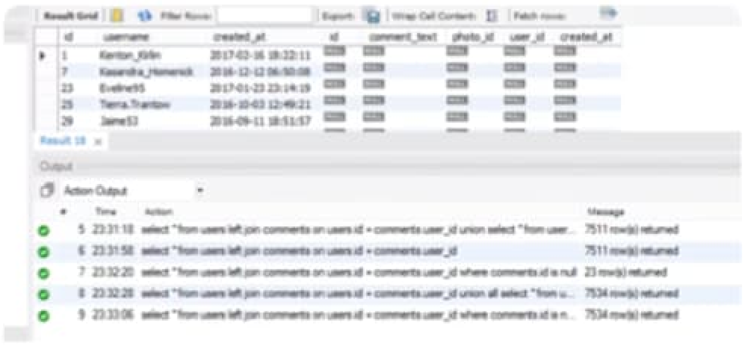
Insights:

Celebrities often have a significant following but may not actively comment. Recognizing users who have never commented helps identify potential celebrities or influencers on the platform

Q.10) Now it's time to find both of them together, find the users who have never commented on any photo or have commented on every photo.

Code:

```
SELECT *
FROM users
LEFT JOIN comments ON users.id = comments.user_id
WHERE comments.id IS NULL
UNION ALL
SELECT *
FROM users
LEFT JOIN comments ON users.id =
comments.user_id;
```



id	username	created_at	id	comment_text	photo_id	user_id	created_at
1	Kenton_Kirlin	2017-02-16 18:22:11	NULL	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL	NULL
23	Eveline95	2017-01-23 23:14:19	NULL	NULL	NULL	NULL	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL	NULL
29	Jaime53	2016-09-11 18:51:57	NULL	NULL	NULL	NULL	NULL

Result 18

Output

Action Output

```
#  Time      Action                                                                                        Message
5  23:31:18  select * from users left join comments on users.id = comments.user_id union select * from user...  7511 row(s) returned
6  23:31:58  select * from users left join comments on users.id = comments.user_id                              7511 row(s) returned
7  23:32:20  select * from users left join comments on users.id = comments.user_id where comments.id is null   23 row(s) returned
8  23:32:28  select * from users left join comments on users.id = comments.user_id union all select * from u... 7534 row(s) returned
9  23:33:06  select * from users left join comments on users.id = comments.user_id where comments.id is n...   7534 row(s) returned
```

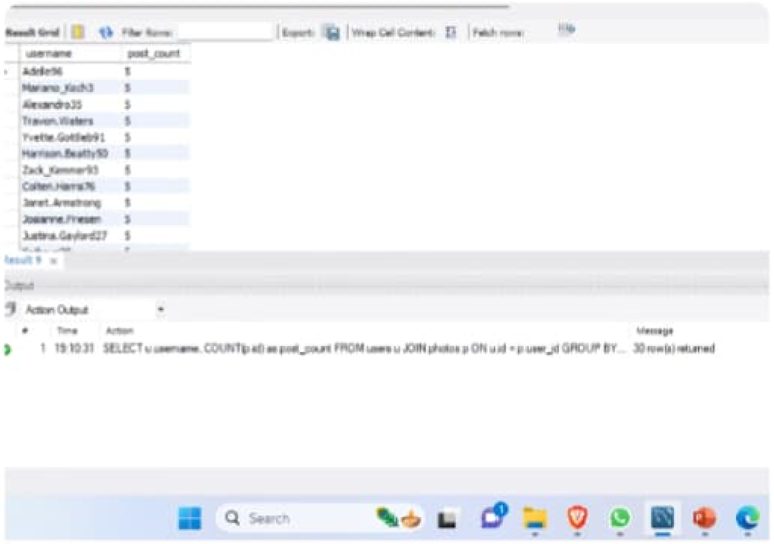
Insights:

Combining users who have never commented with those who have commented on every photo provides a nuanced understanding of user engagement patterns. It allows for targeted communication strategies.

Q.11) Demonstrate the top 30 usernames to the company who have posted photos in the range of 3 to 5.

Code:

```
SELECT u.username, COUNT(p.id) as post_count
FROM users u
JOIN photos p ON u.id = p.user_id
GROUP BY p.user_id
HAVING post_count BETWEEN 3 AND 5
ORDER BY post_count DESC
LIMIT 30;
```



username	post_count
Adelle96	5
Mariano_Koch3	5
Alexandro35	5
Travon.Waters	5
Yvette.Gottlieb91	5
Harrison.Beatty50	5
Zack_Kemmer93	5
Colten.Harris76	5
Janet.Armstrong	5
Josianne.Friesen	5
Justina.Gaylord27	5

```
#   Time      Action                                                                                              Message
1   19:10:31  SELECT u.username, COUNT(p.id) as post_count FROM users u JOIN photos p ON u.id = p.user_id GROUP BY...  30 row(s) returned
```

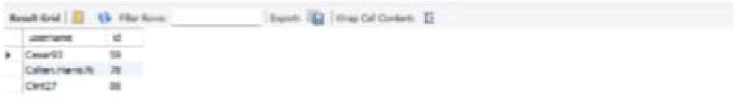
Insights:

The query identifies users posting 3 to 5 photos, emphasizing their diverse contributions for community building. Visual aids enhance engagement metrics, contributing to a healthier platform. Insights also guide user recognition initiatives, fostering a more inclusive community.

Q.12) Can you help me find the users whose name starts with c and ends with any number and have posted the photos as well as liked the photos?

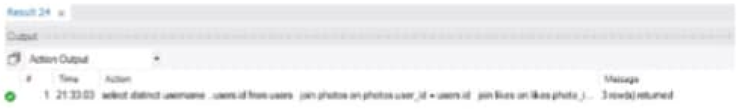
Code:

```
SELECT DISTINCT username, users.id
FROM users
JOIN photos ON photos.user_id = users.id
JOIN likes ON likes.photo_id = photos.id
WHERE username REGEXP '^c' AND username
REGEXP '[0-9]$';
```



username	id
Cesar93	59
Colten.Harris76	78
Clint27	88

Insights:



```
#   Time      Action                                                                                                   Message
1   21:33:03  select distinct username , users.id from users  join photos on photos.user_id = users.id  join likes on likes.photo_i...  3 row(s) returned
```

Identify users with names starting with 'c' and ending with a number who have both posted photos and liked others' photos. This query helps pinpoint engaged users with specific username patterns, providing insights for targeted engagement strategies or user recognition programs.

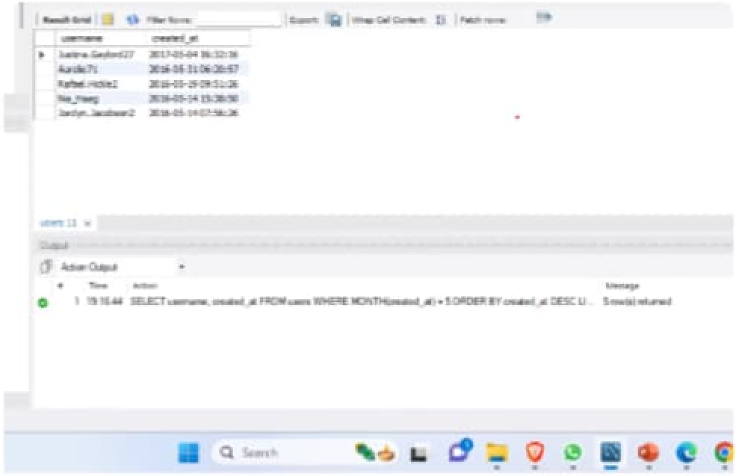
Q.13) Find the users who have created instagramid in may and select top 5 newest joinees from it?

Code:

```
SELECT username, created_at
FROM users
WHERE MONTH(created_at) = 5
ORDER BY created_at DESC
LIMIT 5;
```

Insights:

Identify users who joined in May, emphasizing the top 5 newest members. Showcase growth trends, highlight platform appeal in specific months, and consider targeted onboarding or engagement strategies for May joiners.



username	created_at
Justina.Gaylord27	2017-05-04 16:32:16
Aurelie71	2016-05-31 06:20:57
Rafael.Hickle2	2016-05-29 09:51:26
Nia_Haag	2016-05-14 15:38:50
Jaclyn_Jacobson2	2016-05-14 07:56:26

```
#   Time      Action                                                                                      Message
1   19:15:44  SELECT username, created_at FROM users WHERE MONTH(created_at) = 5 ORDER BY created_at DESC L...  5 row(s) returned
```

Thank You