TRAINITY PROJECT 2

Instagram User Analytics

DESCRIPTION

This project aims to analyze user activity, engagement, and growth on Instagram using SQL to perform various queries on a relational database containing information about users, posts, and more. The analysis will help derive insights into user behaviour, content performance.

APPROACH

I used the given database and tasks to write a mysql query based on my knowledge to get the required output

TECH-STACK USED

I used the mysql workbench 8.0 CE software to write the sql code. I chose it because of my knowledge in the software and is easy to use

INSIGTHS

various insights can be drawn from the data to guide decision-making, optimize social media strategies, and improve user engagement. Here are some potential **insights** you could derive from the analysis:

1. User Engagement Insights

2. Top Content Performance

3. Follower Growth

4. Hashtag Effectiveness

RESULTS

The results from this project are from the outputs gotten from the tasks given in the project

By doing this project I have gotten a stronger grip on my sql skills and my concepts are more clear

The analysis done in this project helps identify various insights or data of a particular information.

A) Marketing Analysis:

1. **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.
   Your Task: Identify the five oldest users on Instagram from the provided database.

CODE:

SELECT

*

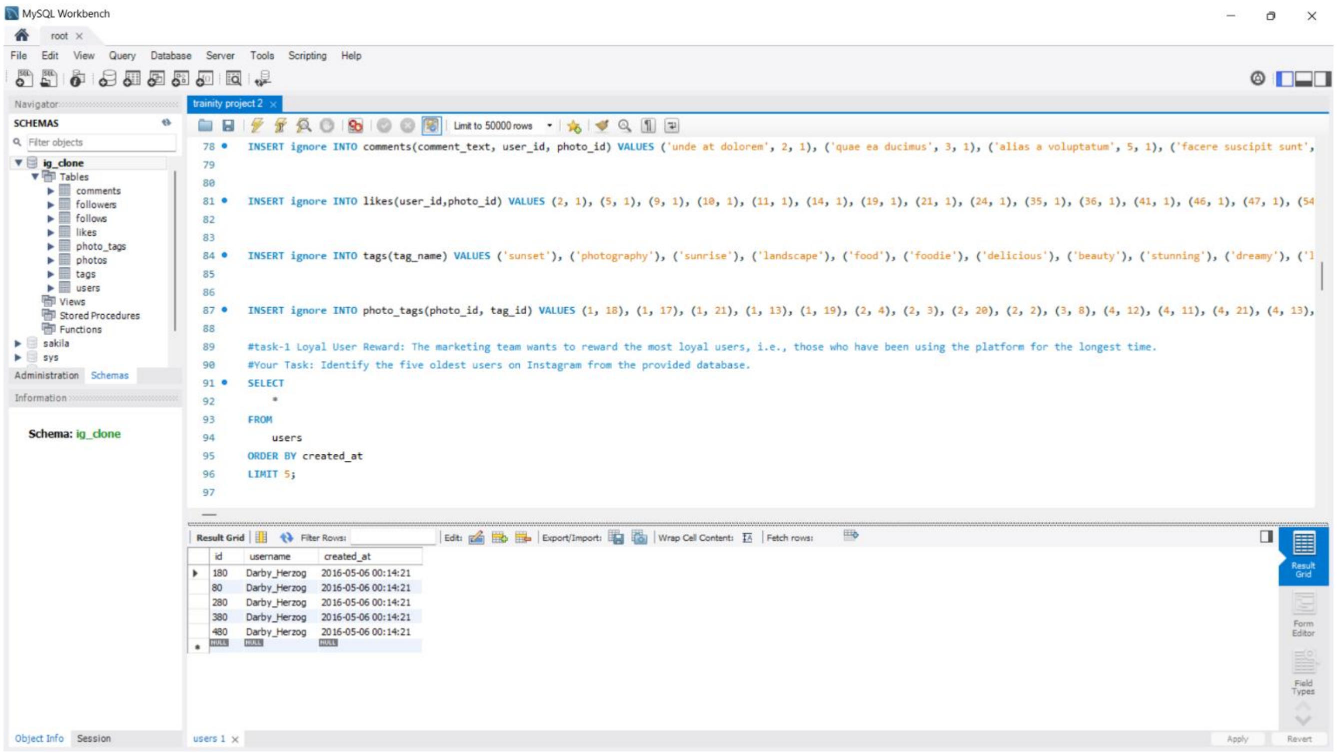
FROM

users

ORDER BY created_at

LIMIT 5;

OUTPUT:



```
INSERT ignore INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolorem', 2, 1), ('quae ea ducimus', 3, 1), ('alias a voluptatum', 5, 1), ('facere suscipit sunt',

INSERT ignore INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (35, 1), (36, 1), (41, 1), (46, 1), (47, 1), (54

INSERT ignore INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicious'), ('beauty'), ('stunning'), ('dreamy'), ('l

INSERT ignore INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2, 2), (3, 8), (4, 12), (4, 11), (4, 21), (4, 13),

#task-1 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.
#Your Task: Identify the five oldest users on Instagram from the provided database.
SELECT
    *
FROM
    users
ORDER BY created_at
LIMIT 5;
```

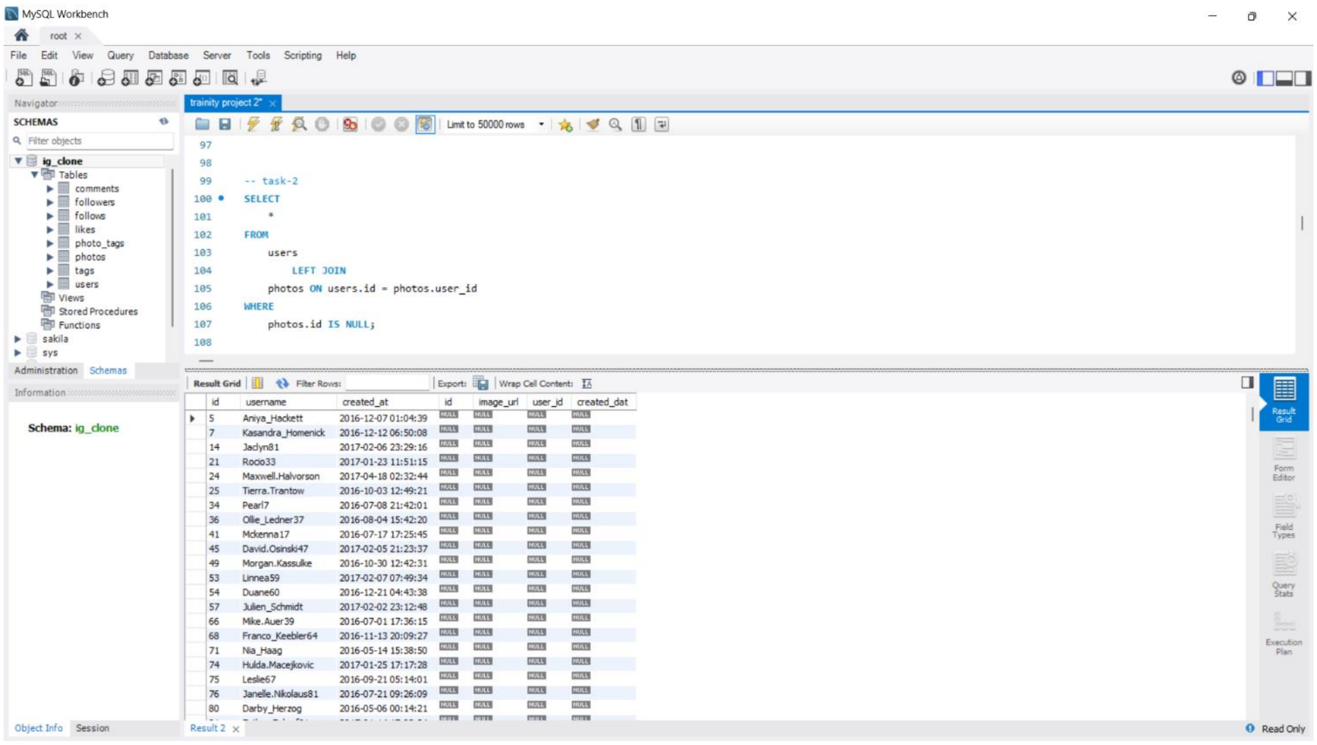
```
id    username       created_at
180   Darby_Herzog   2016-05-06 00:14:21
80    Darby_Herzog   2016-05-06 00:14:21
280   Darby_Herzog   2016-05-06 00:14:21
380   Darby_Herzog   2016-05-06 00:14:21
480   Darby_Herzog   2016-05-06 00:14:21
NULL  NULL           NULL
```

2. **Inactive User Engagement:** The team wants to encourage inactive users to start posting by sending them promotional emails.
   Your Task: Identify users who have never posted a single photo on Instagram.

CODE:

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

OUTPUT:



```
-- task-2
SELECT
    *
FROM
    users
        LEFT JOIN
    photos ON users.id = photos.user_id
WHERE
    photos.id IS NULL;
```

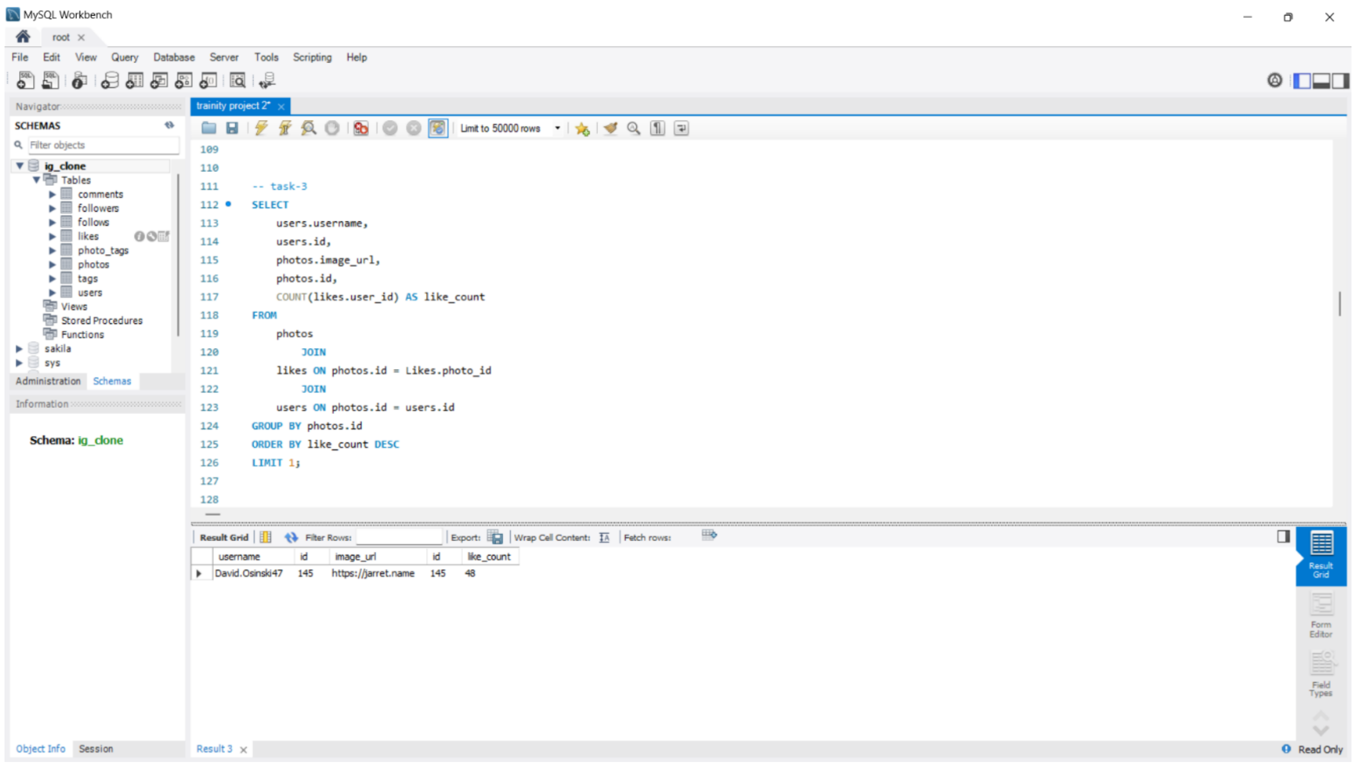
id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jaclyn81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
21	Rocio33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
24	Maxwell.Halvorson	2017-04-18 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

3. **Contest Winner Declaration:** 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.

CODE:

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

OUTPUT:



```
-- task-3
SELECT
    users.username,
    users.id,
    photos.image_url,
    photos.id,
    COUNT(likes.user_id) AS like_count
FROM
    photos
        JOIN
    likes ON photos.id = Likes.photo_id
        JOIN
    users ON photos.id = users.id
GROUP BY photos.id
ORDER BY like_count DESC
LIMIT 1;
```

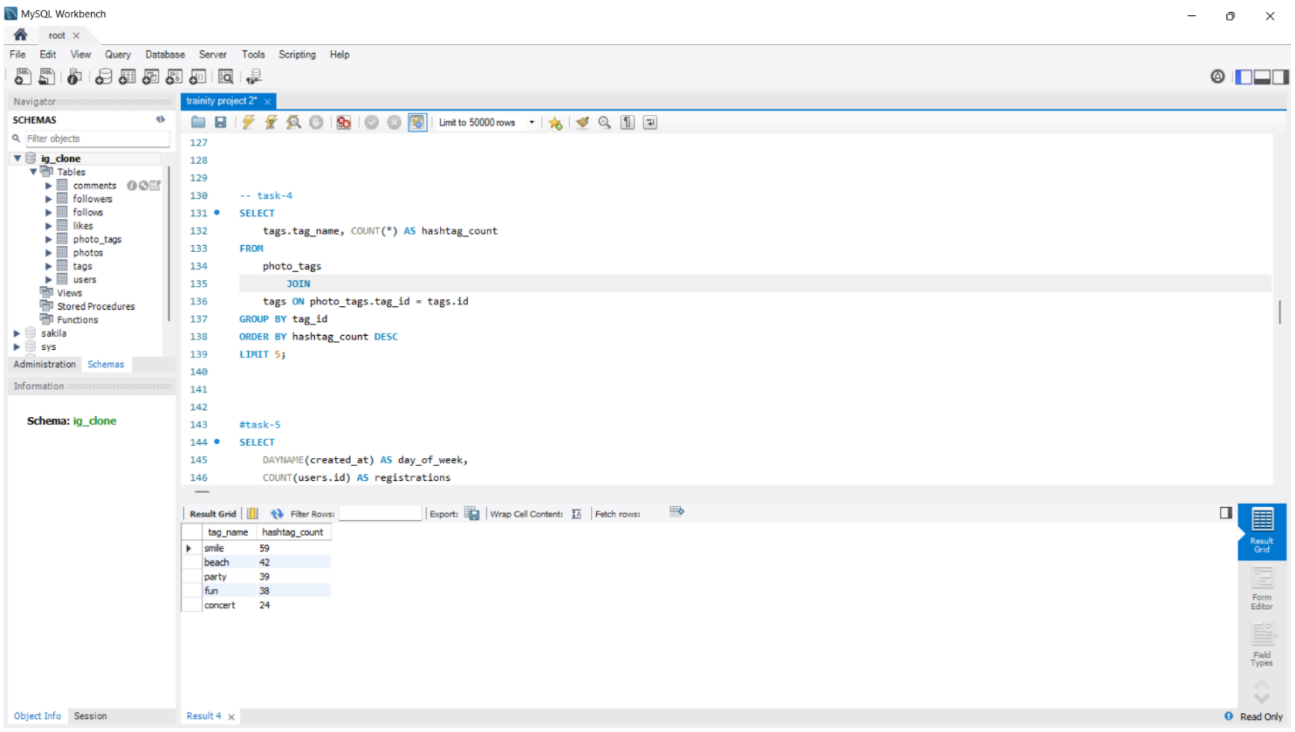
username	id	image_url	id	like_count
David.Osinski47	145	https://jarret.name	145	48

4. **Hashtag Research:** A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
   Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

CODE:

```
SELECT

   tags.tag_name, COUNT(*) AS hashtag_count

FROM

   photo_tags

      JOIN

   tags ON photo_tags.tag_id = tags.id

GROUP BY tag_id

ORDER BY hashtag_count DESC

LIMIT 5;
```

OUTPUT:



```
-- task-4
SELECT
    tags.tag_name, COUNT(*) AS hashtag_count
FROM
    photo_tags
        JOIN
    tags ON photo_tags.tag_id = tags.id
GROUP BY tag_id
ORDER BY hashtag_count DESC
LIMIT 5;


#task-5
SELECT
    DAYNAME(created_at) AS day_of_week,
    COUNT(users.id) AS registrations
```

```
tag_name  hashtag_count
smile     59
beach     42
party     39
fun       38
concert   24
```

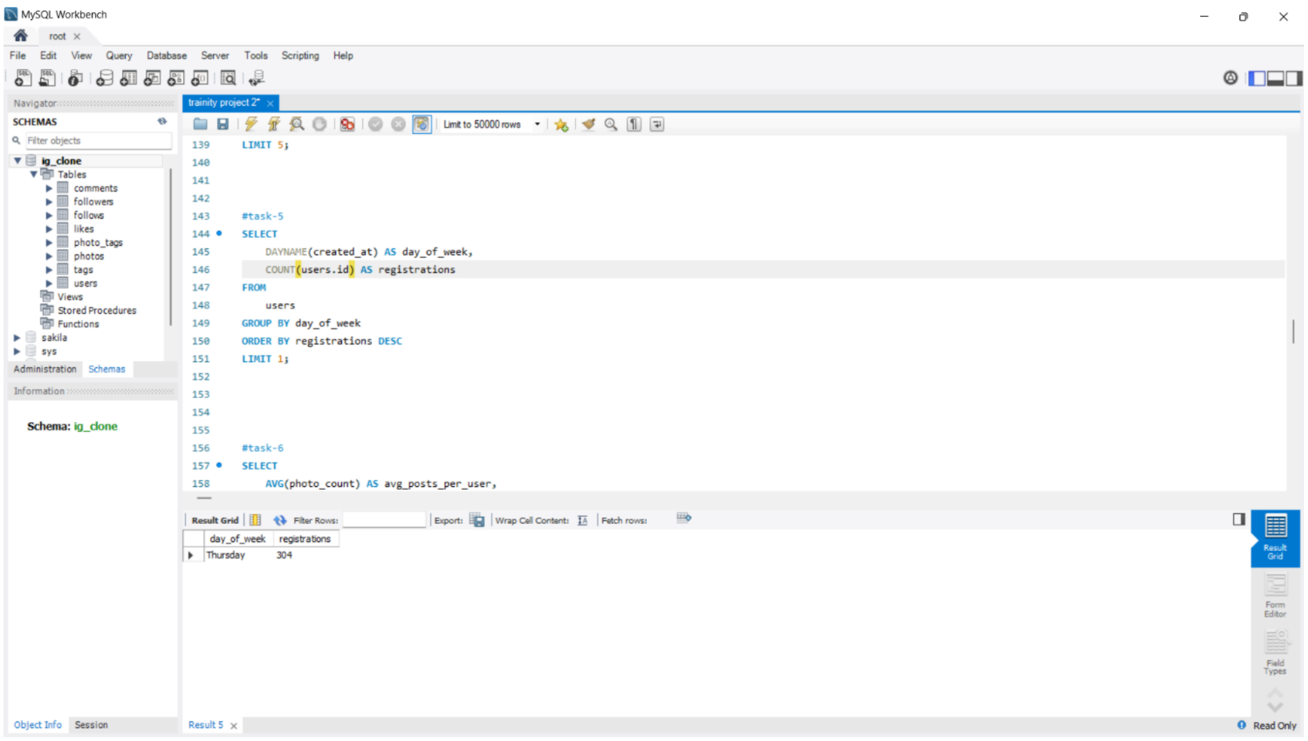
5. **Ad Campaign Launch:** The team wants to know the best day of the week to launch ads.
   Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

CODE:

```
SELECT
```

```
    DAYNAME(created_at) AS day_of_week,
    COUNT(users.id) AS registrations
FROM
    users
GROUP BY day_of_week
ORDER BY registrations DESC
LIMIT 1;
```

OUTPUT:



```
LIMIT 5;



#task-5
SELECT
    DAYNAME(created_at) AS day_of_week,
    COUNT(users.id) AS registrations
FROM
    users
GROUP BY day_of_week
ORDER BY registrations DESC
LIMIT 1;




#task-6
SELECT
    AVG(photo_count) AS avg_posts_per_user,
```

day_of_week	registrations
Thursday	304

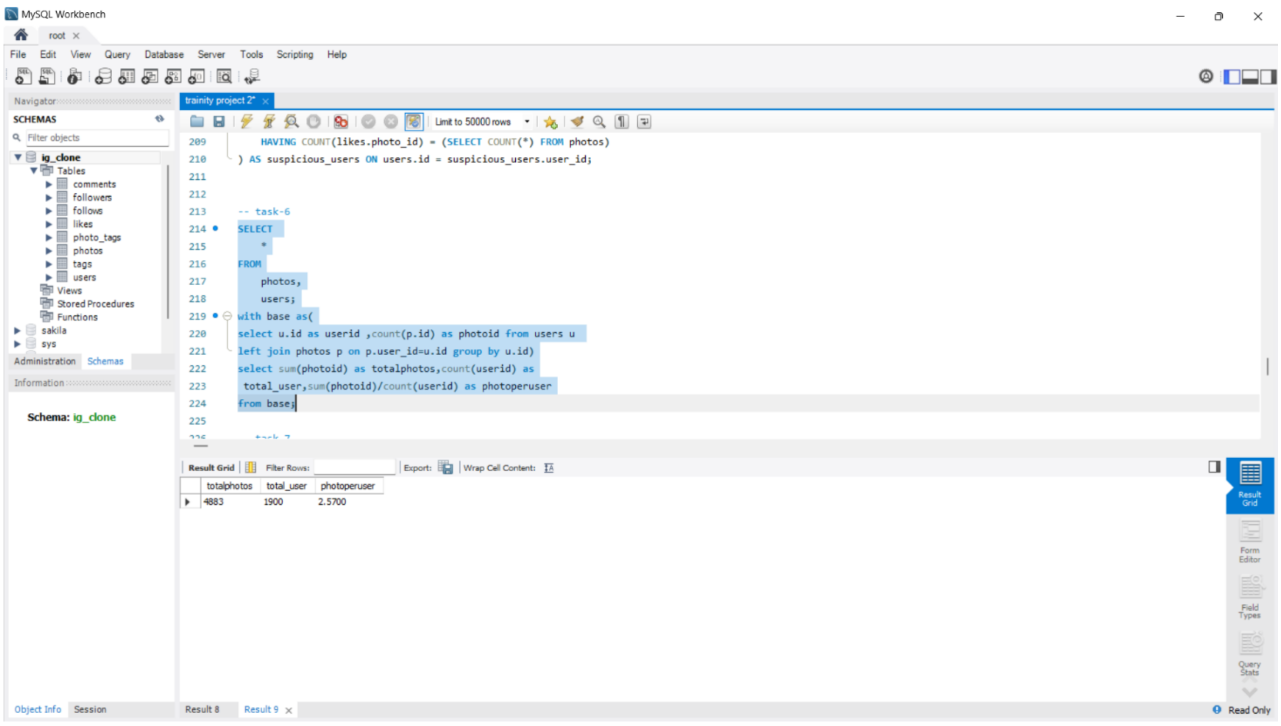
B) Investor Metrics:

1. **User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
   Your 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.

CODE:

```
SELECT
    *
FROM
    photos,
    users;
with base as(
select u.id as userid ,count(p.id) as photoid from users u
left join photos p on p.user_id=u.id group by u.id)
select sum(photoid) as totalphotos,count(userid) as
 total_user,sum(photoid)/count(userid) as photoperuser
from base;
```

OUTPUT:



```
        HAVING COUNT(likes.photo_id) = (SELECT COUNT(*) FROM photos)
    ) AS suspicious_users ON users.id = suspicious_users.user_id;


-- task-6
SELECT
    *
FROM
    photos,
    users;
with base as(
select u.id as userid ,count(p.id) as photoid from users u
left join photos p on p.user_id=u.id group by u.id)
select sum(photoid) as totalphotos,count(userid) as
 total_user,sum(photoid)/count(userid) as photoperuser
from base;
```

```
totalphotos  total_user  photoperuser
4883         1900        2.5700
```

2. **Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts.
   Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

CODE:

```
SELECT
    *
FROM
    users,
    likes;
with base as(
```

select u.username,count(l.photoid) as likes from likes l

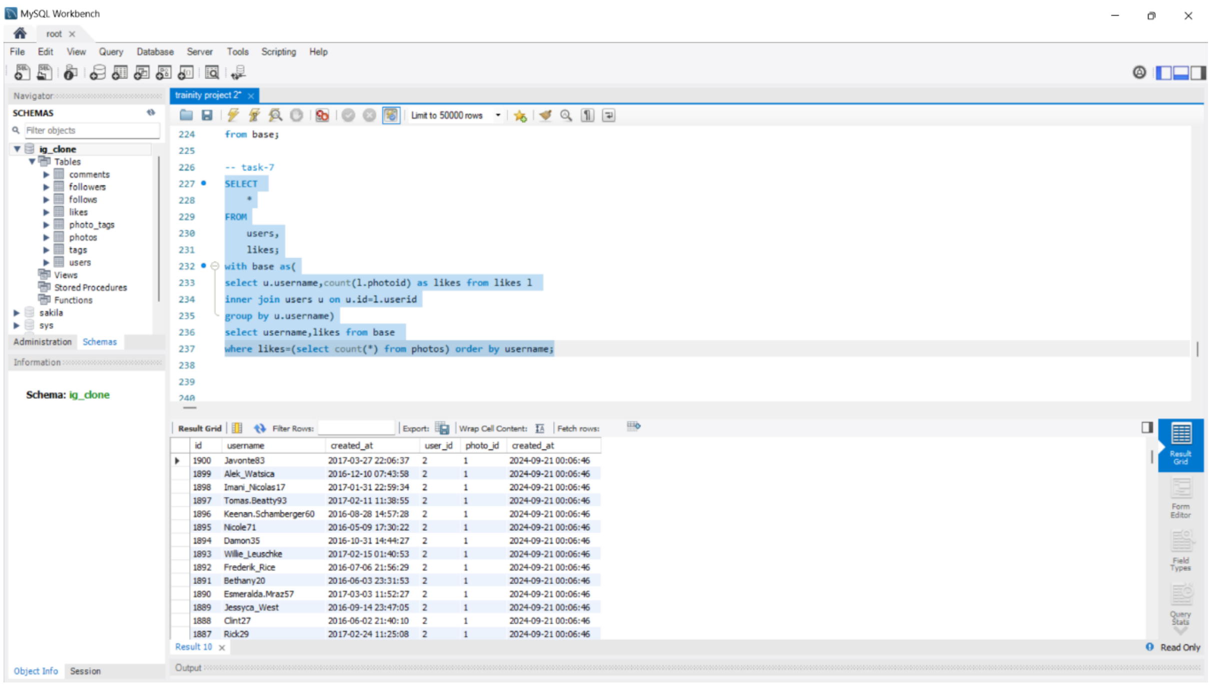
inner join users u on u.id=l.userid

group by u.username)

select username,likes from base

where likes=(select count(*) from photos) order by username;

OUTPUT:



```
from base;

-- task-7
SELECT
    *
FROM
    users,
    likes;
with base as(
select u.username,count(l.photoid) as likes from likes l
inner join users u on u.id=l.userid
group by u.username)
select username,likes from base
where likes=(select count(*) from photos) order by username;
```

id	username	created_at	user_id	photo_id	created_at
1900	Javonte83	2017-03-27 22:06:37	2	1	2024-09-21 00:06:46
1899	Alek_Watsica	2016-12-10 07:43:58	2	1	2024-09-21 00:06:46
1898	Imani_Nicolas17	2017-01-31 22:59:34	2	1	2024-09-21 00:06:46
1897	Tomas.Beatty93	2017-02-11 11:38:55	2	1	2024-09-21 00:06:46
1896	Keenan.Schamberger60	2016-08-28 14:57:28	2	1	2024-09-21 00:06:46
1895	Nicole71	2016-05-09 17:30:22	2	1	2024-09-21 00:06:46
1894	Damon35	2016-10-31 14:44:27	2	1	2024-09-21 00:06:46
1893	Willie_Leuschke	2017-02-15 01:40:53	2	1	2024-09-21 00:06:46
1892	Frederik_Rice	2016-07-06 21:56:29	2	1	2024-09-21 00:06:46
1891	Bethany20	2016-06-03 23:31:53	2	1	2024-09-21 00:06:46
1890	Esmeralda.Mraz57	2017-03-03 11:52:27	2	1	2024-09-21 00:06:46
1889	Jessyca_West	2016-09-14 23:47:05	2	1	2024-09-21 00:06:46
1888	Clint27	2016-06-02 21:40:10	2	1	2024-09-21 00:06:46
1887	Rick29	2017-02-24 11:25:08	2	1	2024-09-21 00:06:46

THANK YOU