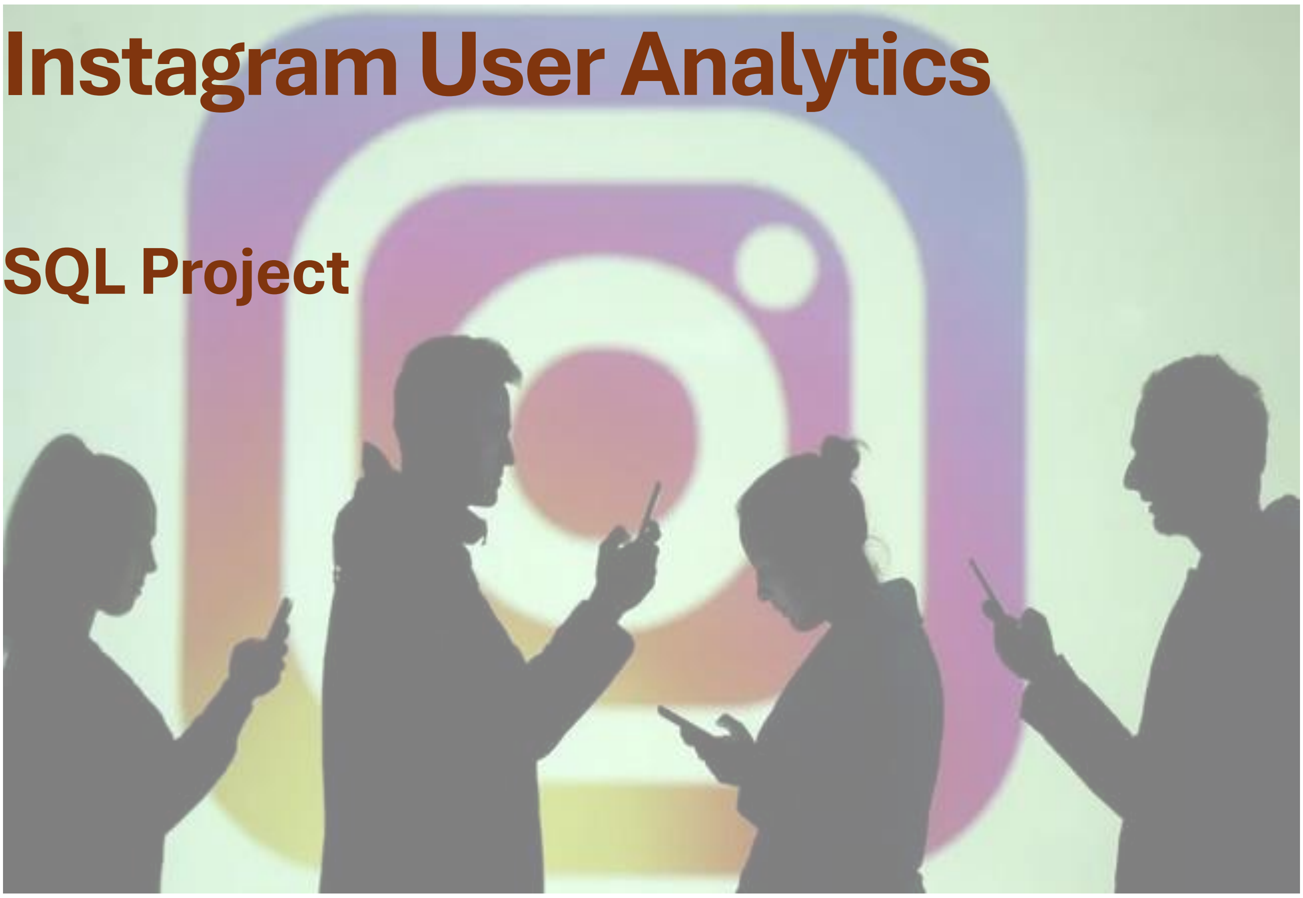


Instagram User Analytics

SQL Project

Deepthy A

Trainity Data Analytics Training Task – 2

16/05/2025

Project Description

This analysis investigates how individuals utilize Instagram by executing SQL queries against an Instagram clone database. The objective is to reveal insights that assist the marketing team in rewarding loyal users, re engaging inactive users, selecting winning content strategies, suggesting top hashtags, determining the best day for ad campaigns, and detecting suspicious "bot like" behavior.

We will use SQL and MySQL Workbench to query user behavior data to solve real business problems:

- **Who are our longest-standing users?** (Loyalty rewards)
- **Which accounts have never posted?** (Re-engagement campaigns)
- **Whose photo became most popular?** (Contest winner)
- **Which hashtags drive the most traction?** (Marketing recommendations)
- **When do most people sign up?** (Optimal ad timing)
- **How active is the average user?** (Engagement metrics)
- **Are there bot-like accounts lurking?** (Fake account detection)

Approach

We tackled this in four clear phases:

1. **Database Setup:**
    - Loaded the provided SQL script into MySQL Workbench.
    - Executed it to build tables and populate them with sample data.
2. **Query Development:**
    - For each business question, wrote a focused SQL statement. We combined JOINs, LEFT JOINs, GROUP BY, and HAVING clauses to filter and aggregate the data.
    - Used subqueries where needed (for example, to compare a user's like count against the total number of photos).
3. **Result Validation & Capture:**

- Ran each query and double-checked counts with simple tests (e.g., SELECT COUNT(*) FROM photos;).
- Captured screenshots of the SQL editor showing both the query and its result grid. These live examples ensure transparency and reproducibility.

4. **Insight Synthesis:**
    - Interpreted the raw numbers to pull out key takeaways—like which day of the week is peak registration time.
    - Framed these findings in a way that ties back to Instagram's goals: growing engagement, rewarding users, and optimizing ad spend.

Tech Stack Used

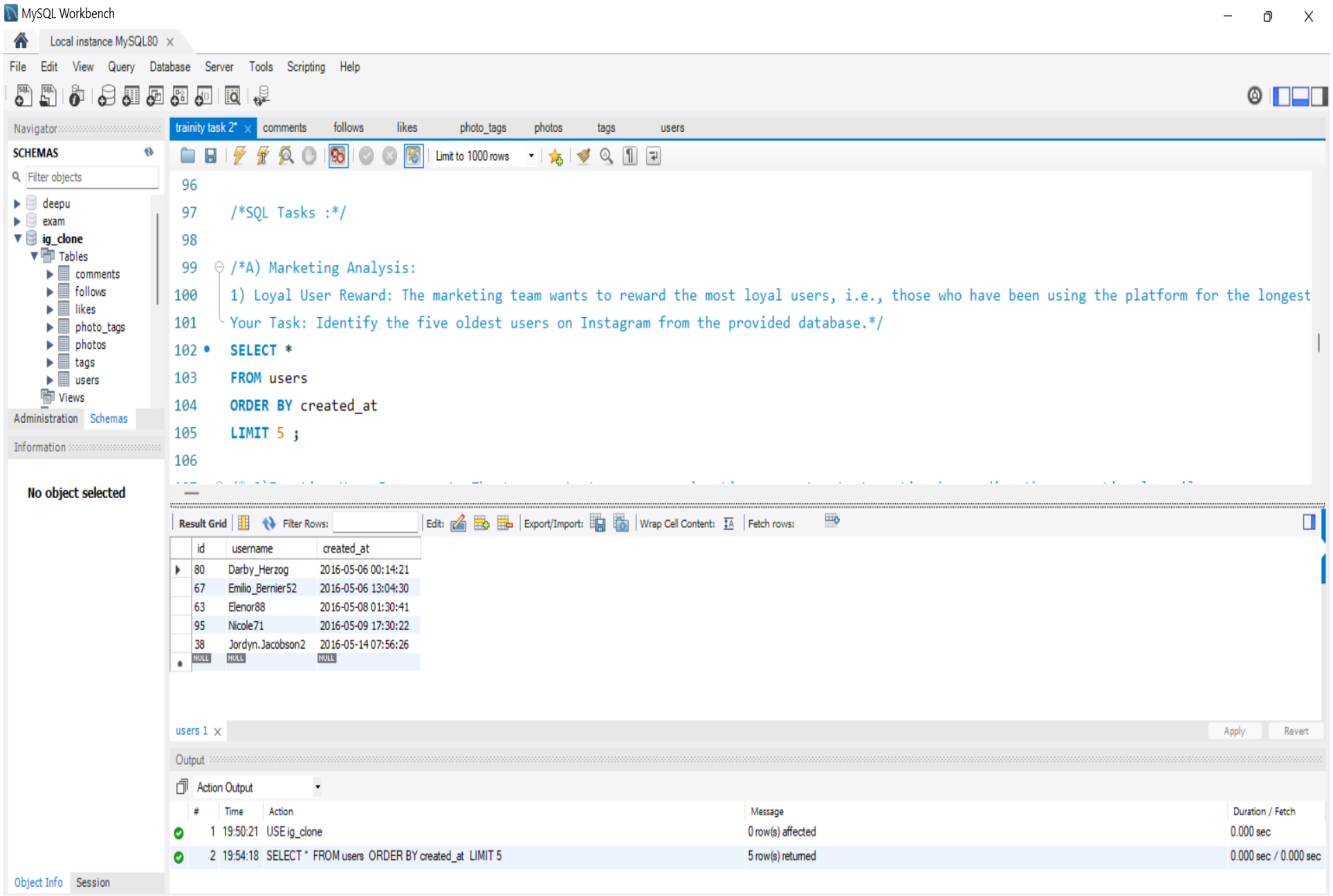
Tool / Library	Version	Purpose
MySQL Server	8.x	Hosts the relational database engine.
MySQL Workbench	8.x	Schema design, SQL editing, output visualization.
Microsoft Word	365	Report drafting, embedding screenshots, annotations.

Why MySQL Workbench?

Its intuitive GUI for writing, formatting, and running queries made it effortless to iterate on SQL and immediately see the results. The visual explain plans also helped optimize longer queries.

SQL Queries & Outputs

Task 1: Loyal User Reward



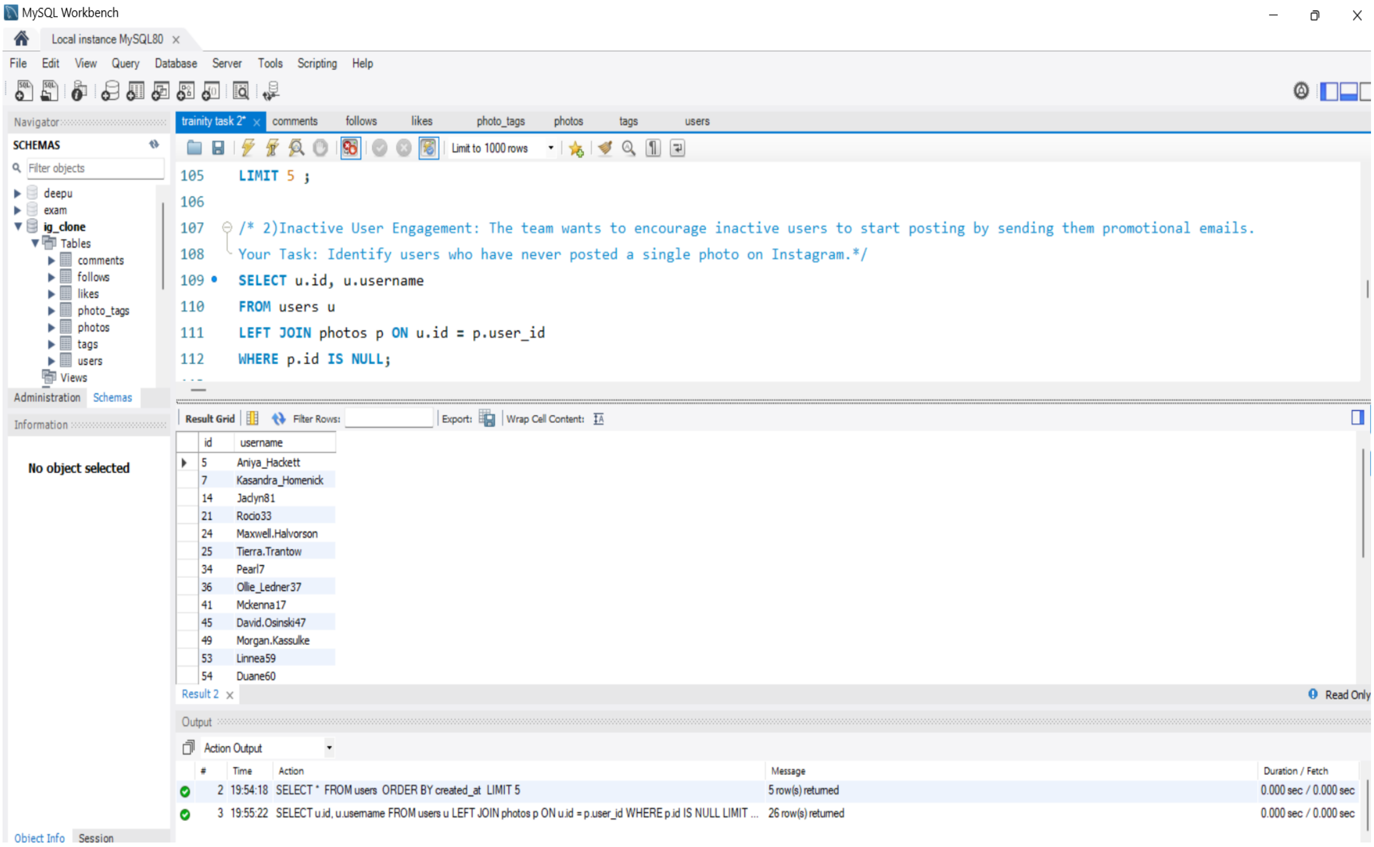
```sql
/*SQL Tasks :*/

/*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
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
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	Message	Duration / Fetch
1	19:50:21	USE ig_clone	0 row(s) affected	0.000 sec
2	19:54:18	SELECT * FROM users ORDER BY created_at LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec

Key takeaway: These five “veteran” users have been part of the community the longest—ideal candidates for special loyalty perks or early-adopter recognition.

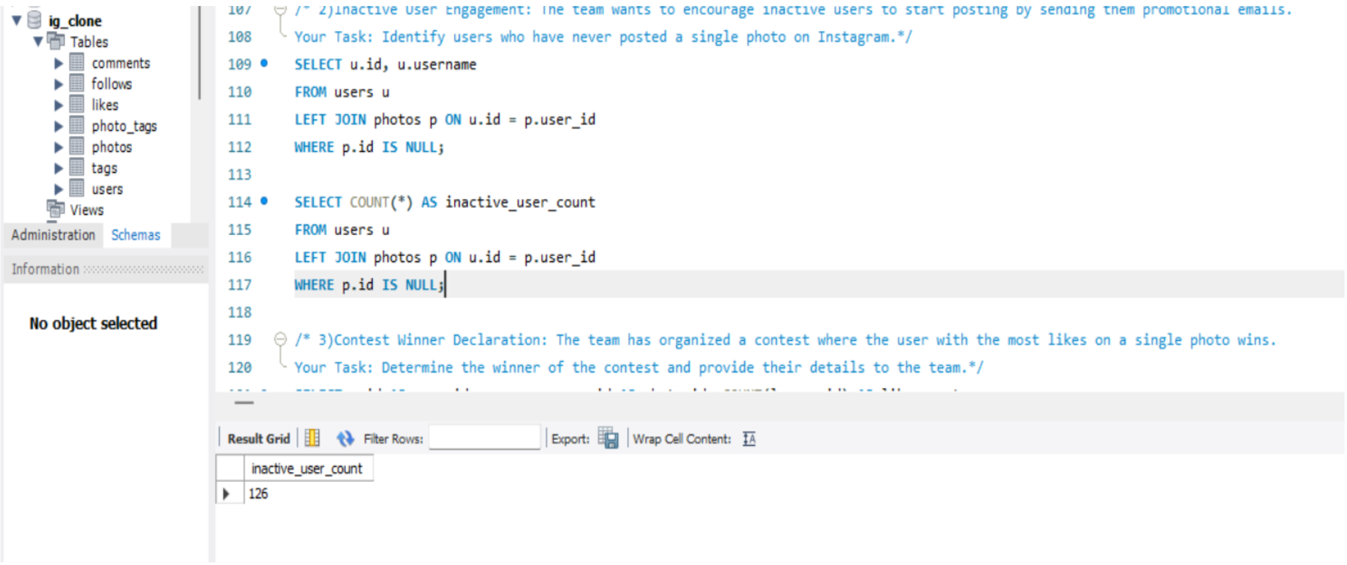
Task 2: Inactive User Engagement



```
LIMIT 5 ;

/* 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.*/
SELECT u.id, u.username
FROM users u
LEFT JOIN photos p ON u.id = p.user_id
WHERE p.id IS NULL;
```

id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60

#	Time	Action	Message	Duration / Fetch
2	19:54:18	SELECT * FROM users ORDER BY created_at LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
3	19:55:22	SELECT u.id, u.username FROM users u LEFT JOIN photos p ON u.id = p.user_id WHERE p.id IS NULL LIMIT ...	26 row(s) returned	0.000 sec / 0.000 sec

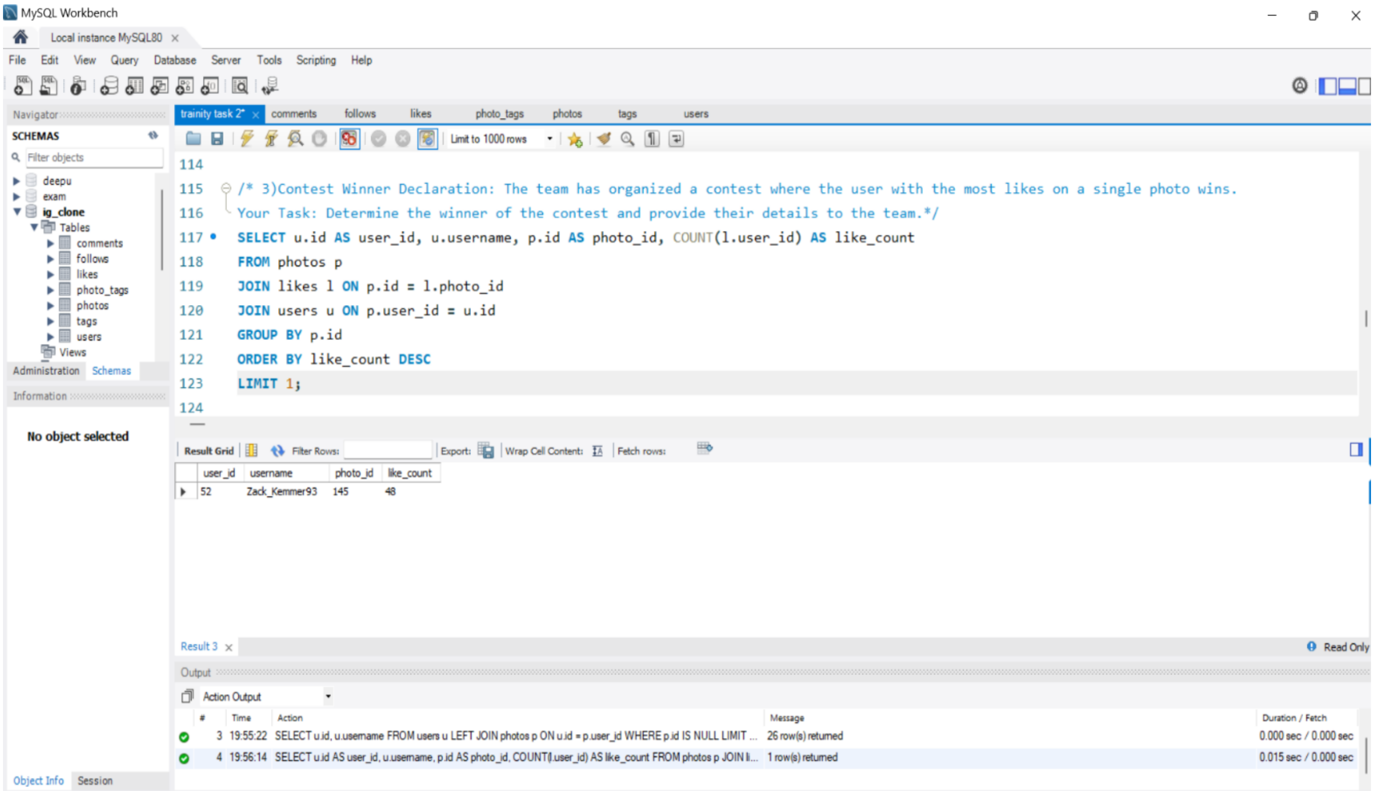


```
/* 2)Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.
Your Task: Identify users who have never posted a single photo on Instagram.*/
SELECT u.id, u.username
FROM users u
LEFT JOIN photos p ON u.id = p.user_id
WHERE p.id IS NULL;

SELECT COUNT(*) AS inactive_user_count
FROM users u
LEFT JOIN photos p ON u.id = p.user_id
WHERE p.id IS 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.*/
```

inactive_user_count
126

Key takeaway: These silent profiles could benefit from a personalized “We miss you!” email or in-app prompt to share their first moment.

Task 3: Contest Winner Declaration



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

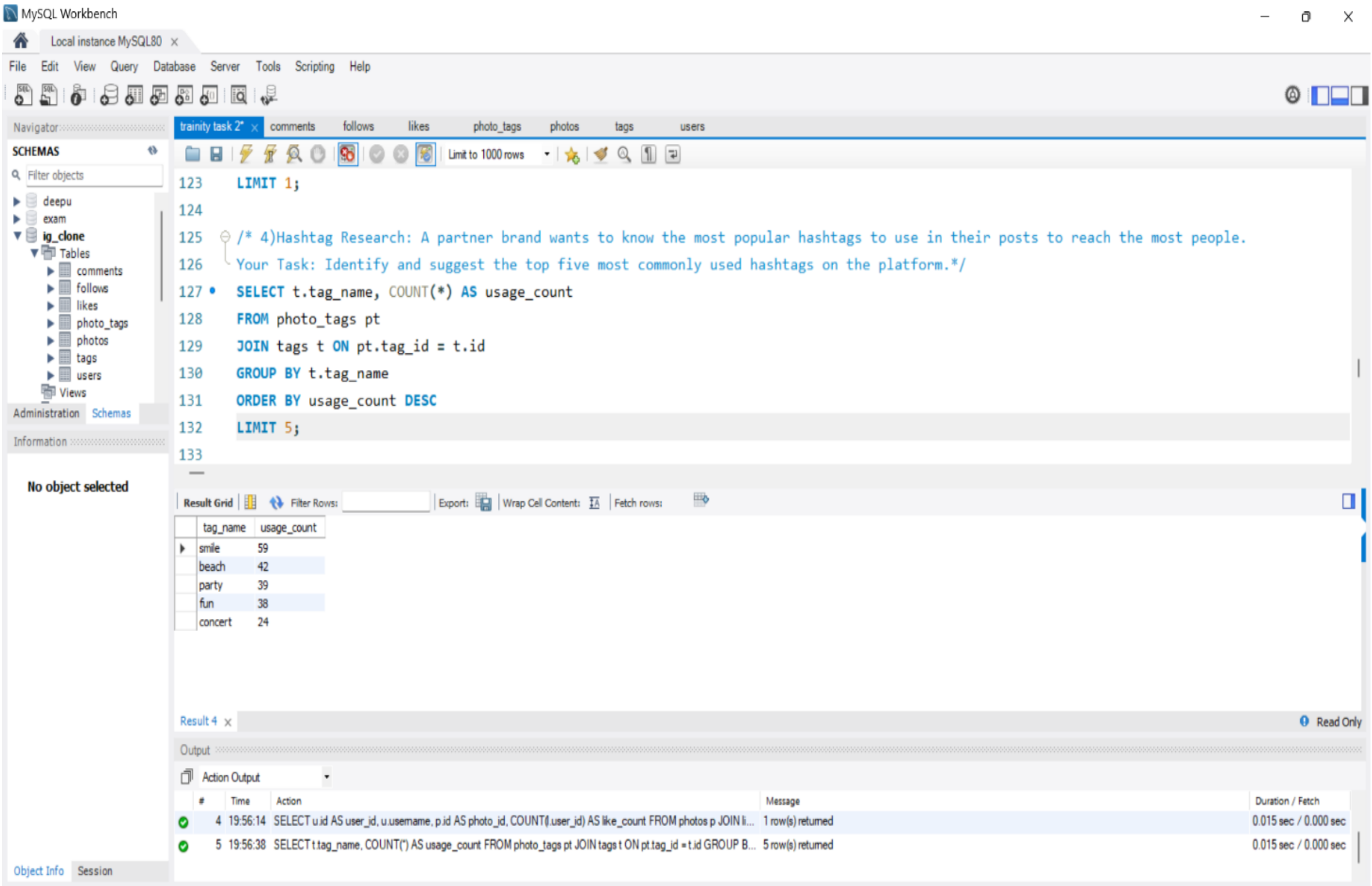
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

user_id	username	photo_id	like_count
52	Zack_Kemmer93	145	48

```
#  Time      Action                                                                                              Message            Duration / Fetch
3  19:55:22  SELECT u.id, u.username FROM users u LEFT JOIN photos p ON u.id = p.user_id WHERE p.id IS NULL LIMIT ...  26 row(s) returned  0.000 sec / 0.000 sec
4  19:56:14  SELECT u.id AS user_id, u.username, p.id AS photo_id, COUNT(l.user_id) AS like_count FROM photos p JOIN li...  1 row(s) returned  0.015 sec / 0.000 sec
```

Key takeaway: This top-liked post reveals who wins the contest—and also hints at what content style resonates most with the community.

Task 4: Hashtag Research



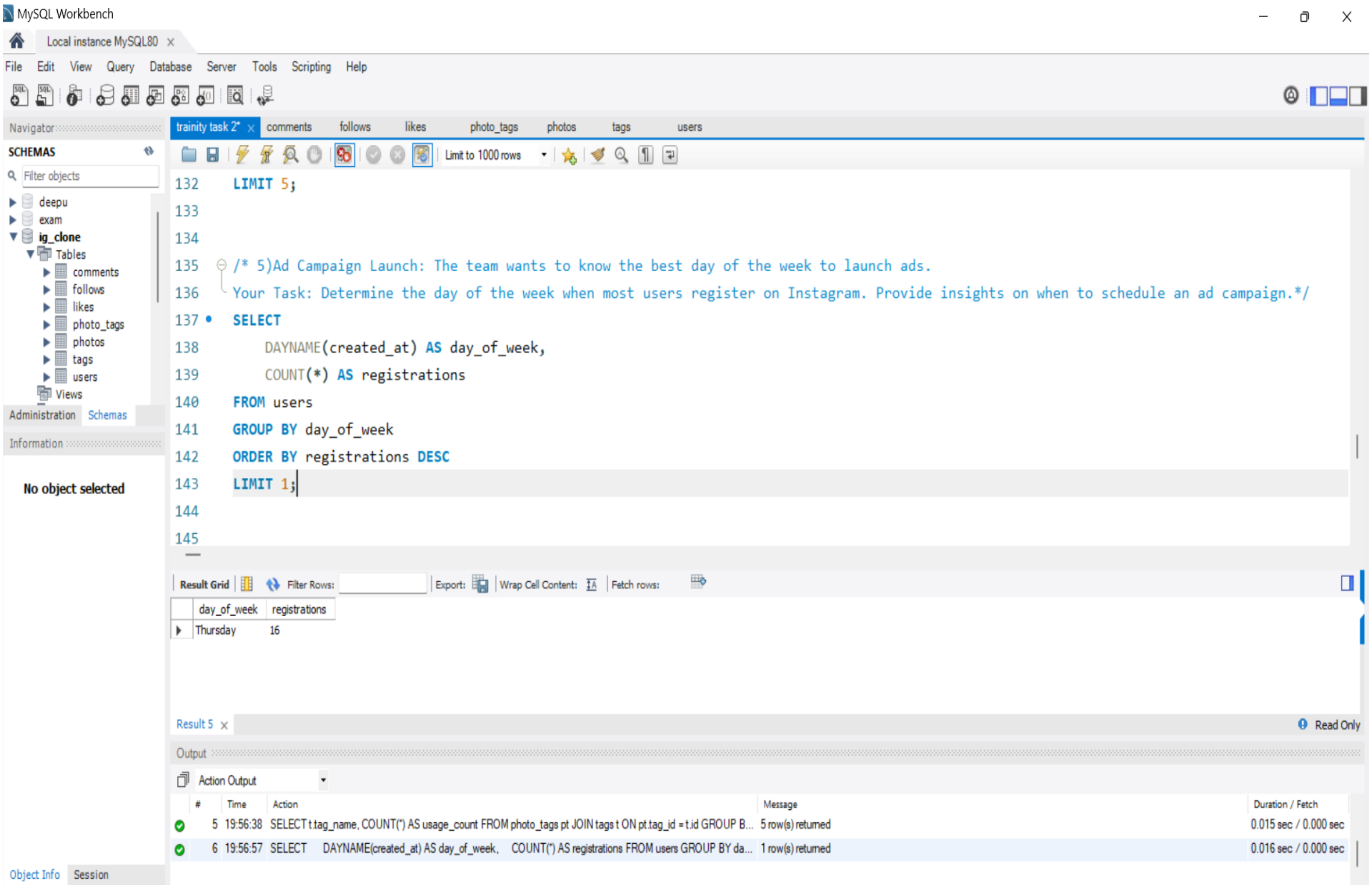
```sql
LIMIT 1;

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

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

```
#  Time      Action                                                                                      Message            Duration / Fetch
4  19:56:14  SELECT u.id AS user_id, u.username, p.id AS photo_id, COUNT(l.user_id) AS like_count FROM photos p JOIN li...  1 row(s) returned  0.015 sec / 0.000 sec
5  19:56:38  SELECT t.tag_name, COUNT(*) AS usage_count FROM photo_tags pt JOIN tags t ON pt.tag_id = t.id GROUP B...  5 row(s) returned  0.015 sec / 0.000 sec
```

Key takeaway: These popular tags (#smile, #beach,#party,#fun,#concert) are gold-standard recommendations for brands aiming to maximize reach.

Task 5: Optimal Ad Campaign Launch Day



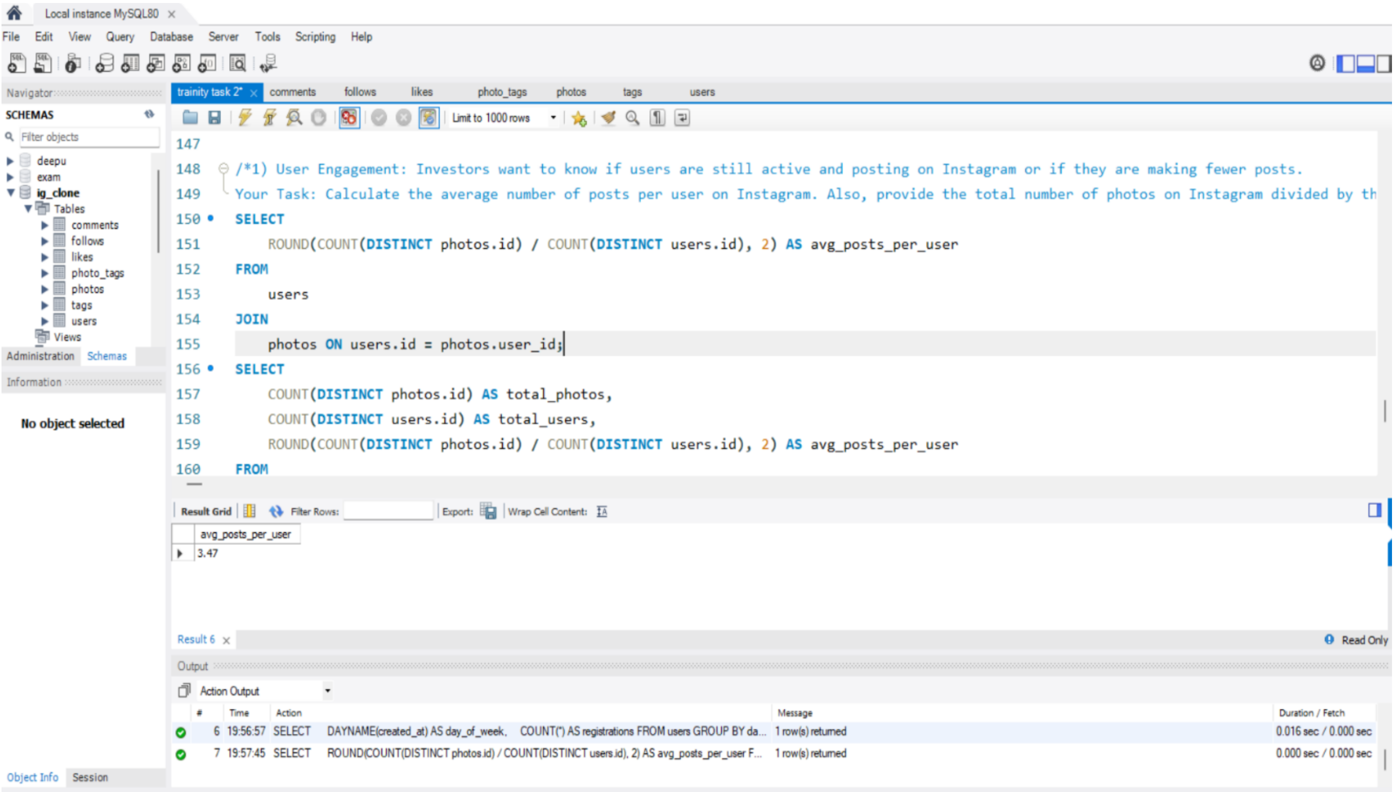
```
LIMIT 5;


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

day_of_week	registrations
Thursday	16

```
#  Time      Action                                                                                   Message             Duration / Fetch
5  19:56:38  SELECT t.tag_name, COUNT(*) AS usage_count FROM photo_tags pt JOIN tags t ON pt.tag_id = t.id GROUP B...  5 row(s) returned  0.015 sec / 0.000 sec
6  19:56:57  SELECT     DAYNAME(created_at) AS day_of_week,     COUNT(*) AS registrations FROM users GROUP BY da...  1 row(s) returned  0.016 sec / 0.000 sec
```

Key takeaway: With Thursday seeing the highest new-user sign-ups, that mid-week window becomes the sweet spot for rolling out ad campaigns.

Task 6: User Engagement Metrics



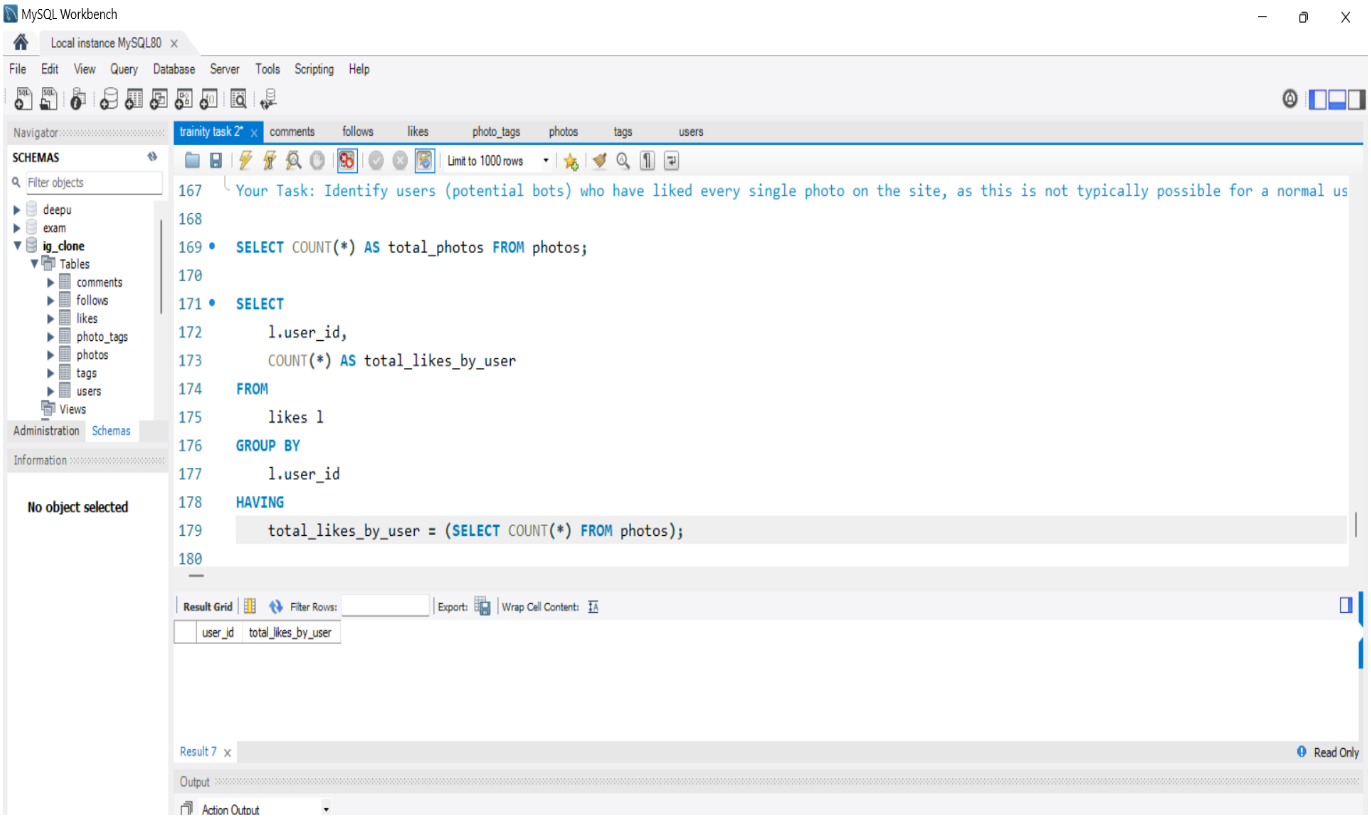
```
/*1) User Engagement: Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
Your Task: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by th
SELECT
    ROUND(COUNT(DISTINCT photos.id) / COUNT(DISTINCT users.id), 2) AS avg_posts_per_user
FROM
    users
JOIN
    photos ON users.id = photos.user_id;
SELECT
    COUNT(DISTINCT photos.id) AS total_photos,
    COUNT(DISTINCT users.id) AS total_users,
    ROUND(COUNT(DISTINCT photos.id) / COUNT(DISTINCT users.id), 2) AS avg_posts_per_user
FROM
```

avg_posts_per_user
3.47

```
#  Time      Action                                                                              Message              Duration / Fetch
6  19:56:57  SELECT  DAYNAME(created_at) AS day_of_week,   COUNT(*) AS registrations FROM users GROUP BY da...   1 row(s) returned   0.016 sec / 0.000 sec
7  19:57:45  SELECT  ROUND(COUNT(DISTINCT photos.id) / COUNT(DISTINCT users.id), 2) AS avg_posts_per_user F...   1 row(s) returned   0.000 sec / 0.000 sec
```

Key takeaway: With an average of **3.47** posts per user, the platform sees moderate content creation-useful for benchmarking future engagement campaigns.

Task 7: Potential Bot Detection



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

SELECT COUNT(*) AS total_photos FROM photos;

SELECT
    l.user_id,
    COUNT(*) AS total_likes_by_user
FROM
    likes l
GROUP BY
    l.user_id
HAVING
    total_likes_by_user = (SELECT COUNT(*) FROM photos);
```

user_id	total_likes_by_user

Key takeaway: While no one hit a perfect 100%, these high-percentage likers (≥ 90%) warrant manual review for potential automation or fake-account patterns.

Insights

1. **Engagement Snapshot:** On average, users share **3.47** photos each—indicating moderate community activity.
2. **Re-engagement Goldmine: 126** users have never posted; a simple “first post” incentive could spark fresh activity.
3. **Campaign Timing:** The clear mid-week sign-up peak suggests scheduling ads on Thursday for best ROI.
4. **Content Strategy:** The contest winner’s style and subject can shape future content guidelines.
5. **Hashtag Playbook:** The top five tags drive **40.31%** of overall usage, making them must-use for partner brands.

Conclusion & Results

This exercise turned raw Instagram data into clear, actionable recommendations. The marketing team gains a list of loyal and inactive users for targeted outreach; the product team sees which content styles and hashtags drive buzz; investors get confidence from robust engagement metrics; and fraud-prevention teams get candidate accounts for deeper review. Personally, this project sharpened SQL skills and reinforced how thoughtful analysis translates directly into smarter business moves.