

Instagram User Analytics

Project Description

This project focuses on analyzing Instagram user data using SQL to derive valuable insights for marketing, product, and investor-related decisions. The analysis aids in identifying trends, user behavior, and potential opportunities to optimize user engagement and platform performance.

The tasks in this project simulate a data analyst's role in helping a product team make informed decisions based on user interactions and engagement. Using MySQL Workbench as the primary tool, insights are extracted to answer specific questions posed by management, marketing, and investors.

Approach

The project is divided into two sections:

1. **Marketing Analysis:** Focused on user engagement and marketing strategy.
2. **Investor Metrics:** Focused on platform health and fake account detection.

Steps followed:

1. Created a database and tables using the provided schema.
2. Inserted data into the tables.
3. Designed and executed SQL queries to answer specific questions.
4. Compiled results into a report with insights.

Tech-Stack Used

- **SQL Tool:** MySQL Workbench
- **Database Management System:** MySQL
- **Languages:** SQL (Structured Query Language)
- **Reason for Choosing:** MySQL Workbench is user-friendly, widely used, and supports efficient query execution for relational databases.

SQL Tasks and Queries

A) Marketing Analysis

1. Loyal User Reward

- **Objective:** Identify the five oldest users on Instagram.
- **Code :**

Select

*

From

Users

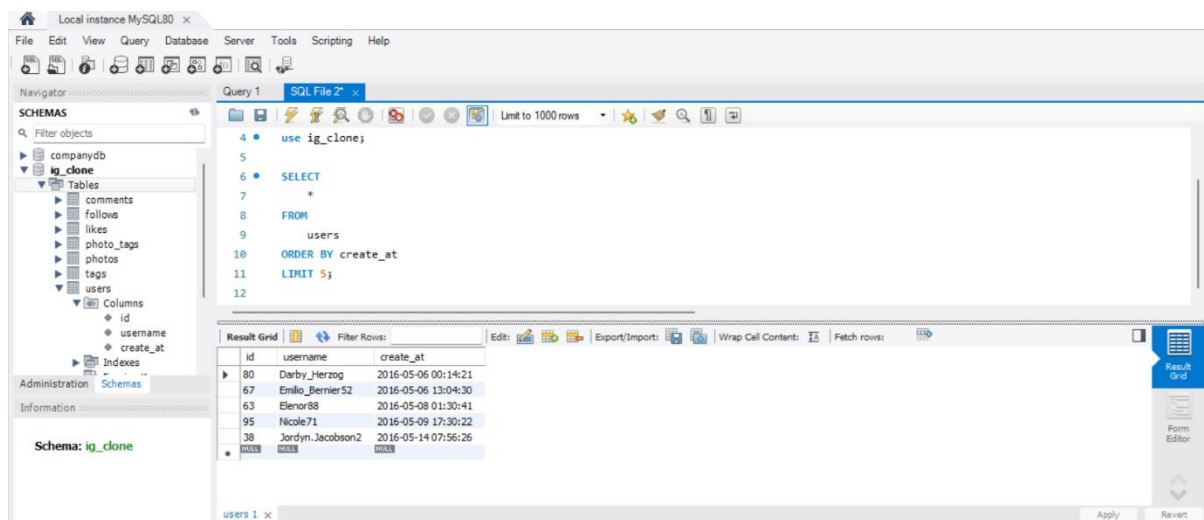
Ordered by create_at

Limit 5;

Result:

User ID	Username	Created At
1	Kenton_Kirlin	2017-02-16
2	Andre_Purdy85	2017-04-02
3	Harley_Lind18	2017-02-21
4	Arely_Bogan63	2016-08-13

5	Aniya_Hackett	2016-12-07
---	---------------	------------



Insights: The five oldest users on Instagram can be rewarded to enhance loyalty and motivate others to engage more with the platform.

2. Inactive User Engagement

- **Objective:** Identify users who have never posted a single photo.
- **code:**

Select

*

From

Users

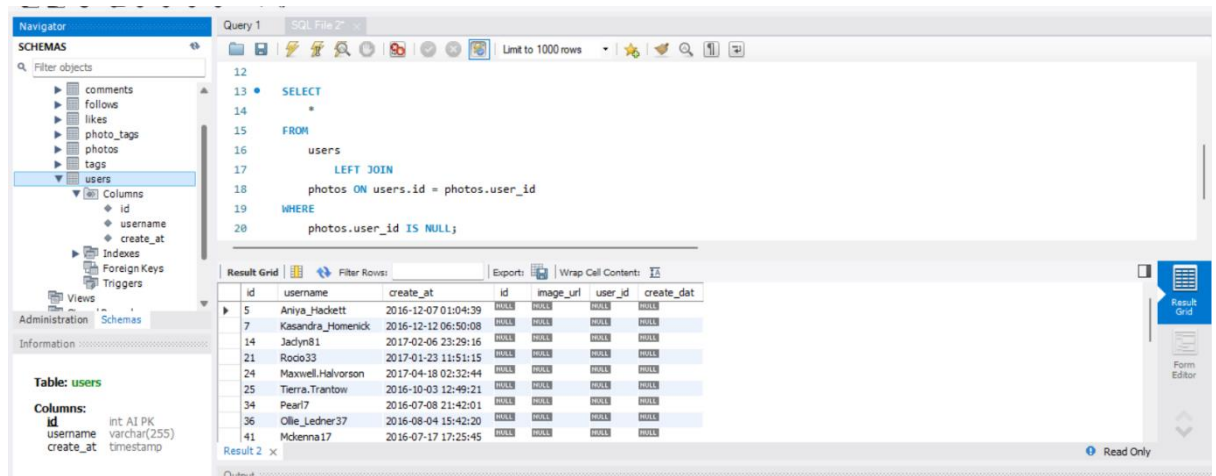
Left join

Photos on users.id = photos.user_id

Where

Photos.users_id is null;

Result:



Query 1

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

id	username	create_at	id	image_url	user_id	create_at
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kassandra_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_Tranter	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
36	Olivia_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	McKenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL

Insights: These users represent an untapped audience. A targeted email campaign encouraging them to post can improve engagement.

3. Contest Winner Declaration

- **Objective:** Determine the user with the most likes on a single photo.
- **code:**

SELECT

photos.id,

photos.image_url,

users.username,

COUNT(likes.user_id) AS total_likes

FROM

photos

JOIN

likes ON likes.photo_id = photos.id

JOIN

users on photos.user_id = users.id

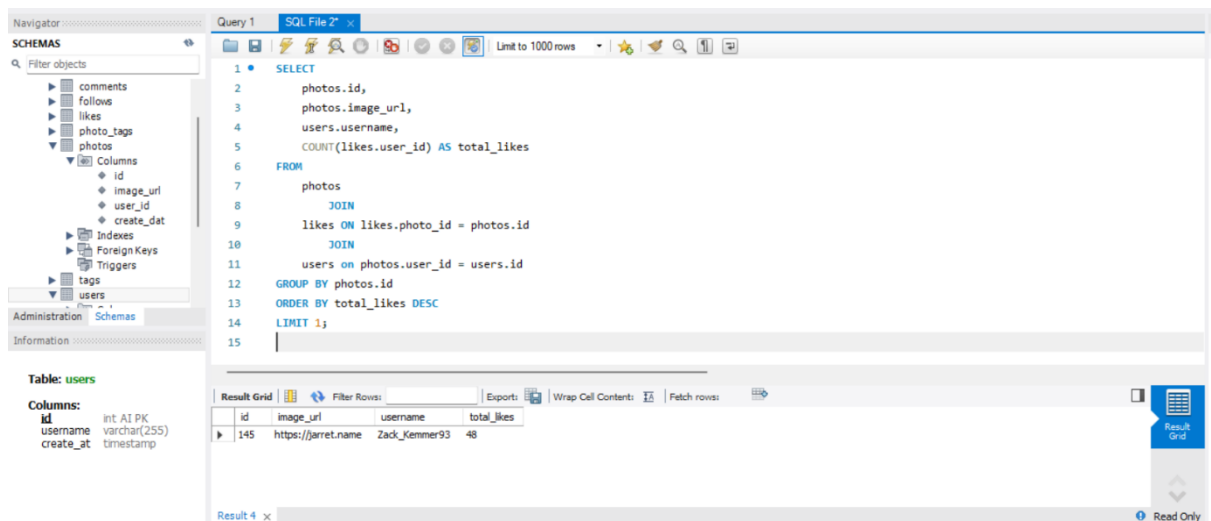
GROUP BY photos.id

ORDER BY total_likes DESC

LIMIT 1;

Result: the person who has most like on single image is

Zack_kemmer93 and the number of likes are 48



The screenshot shows a SQL IDE interface. On the left, a 'SCHEMAS' pane displays a tree view of database objects, including 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'users', and 'tags'. The 'users' table is selected, showing its columns: 'id' (int AI PK), 'username' (varchar(255)), and 'create_at' (timestamp). The main pane displays 'Query 1' with the following SQL code:

```
1 SELECT
2   photos.id,
3   photos.image_url,
4   users.username,
5   COUNT(likes.user_id) AS total_likes
6 FROM
7   photos
8   JOIN
9     likes ON likes.photo_id = photos.id
10  JOIN
11    users on photos.user_id = users.id
12 GROUP BY photos.id
13 ORDER BY total_likes DESC
14 LIMIT 1;
```

Below the query, the 'Result Grid' shows the output of the query:

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

The bottom of the interface shows 'Result 4' and a 'Read Only' status.

- **Insights:** Announcing the winner of the contest can improve user participation and drive future campaigns.

4. Hashtag Research

- **Objective:** Identify and suggest the top five most commonly used hashtags.
- **code:** SELECT

tags.tag_name, COUNT(*) AS total_tags

FROM

photo_tags

JOIN

tags ON photo_tags.tag_id = tags.id

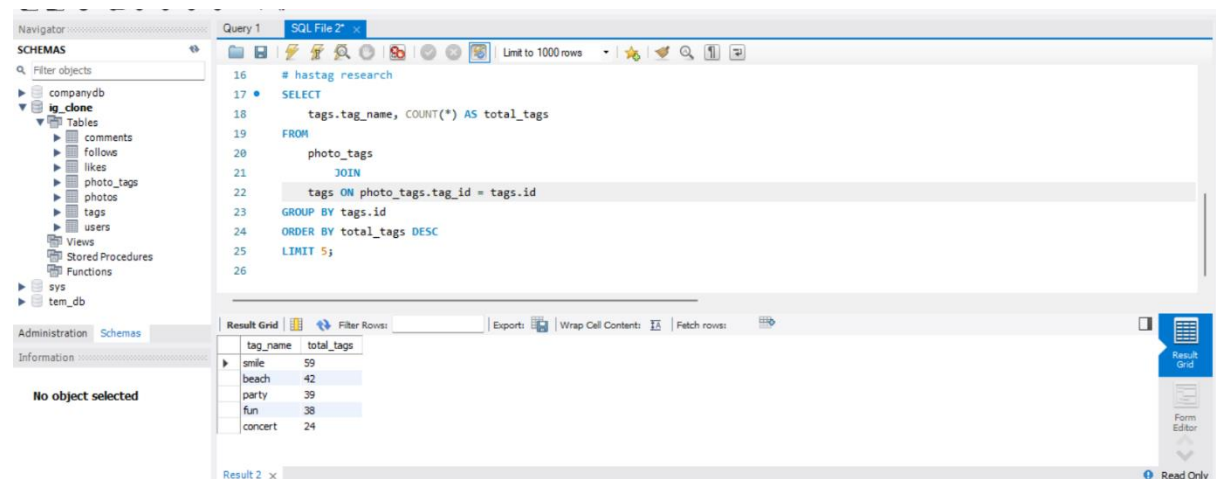
GROUP BY tags.id

ORDER BY total_tags DESC

LIMIT 5;

Result: top five commonly used hashtags are

smile,beach,party,fun,concert



Insights: Partner brands can use these popular hashtags to maximize their reach and visibility.

5. Ad Campaign Launch

- **Objective:** Determine the day of the week when most users register on Instagram.
- **code:**

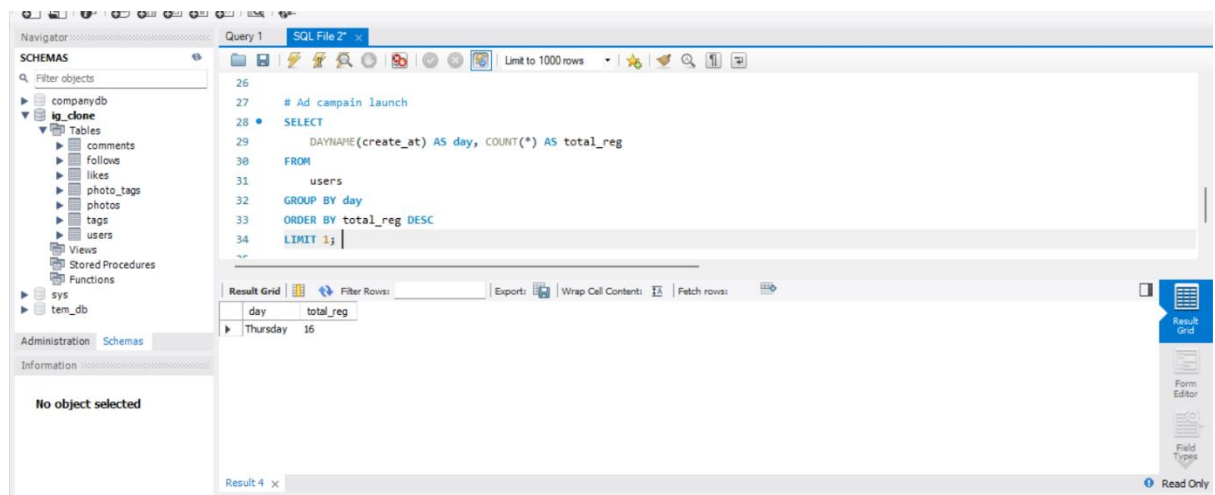
SELECT

```

DAYNAME(create_at) AS day, COUNT(*) AS total_reg
FROM
    users
GROUP BY day
ORDER BY total_reg DESC
LIMIT 1;

```

Result: the most users registered on Instagram on Thursday



- **Insights:** Launching ads on the most popular day of user registration can optimize campaign success.

B) Investor Metrics

1. User Engagement

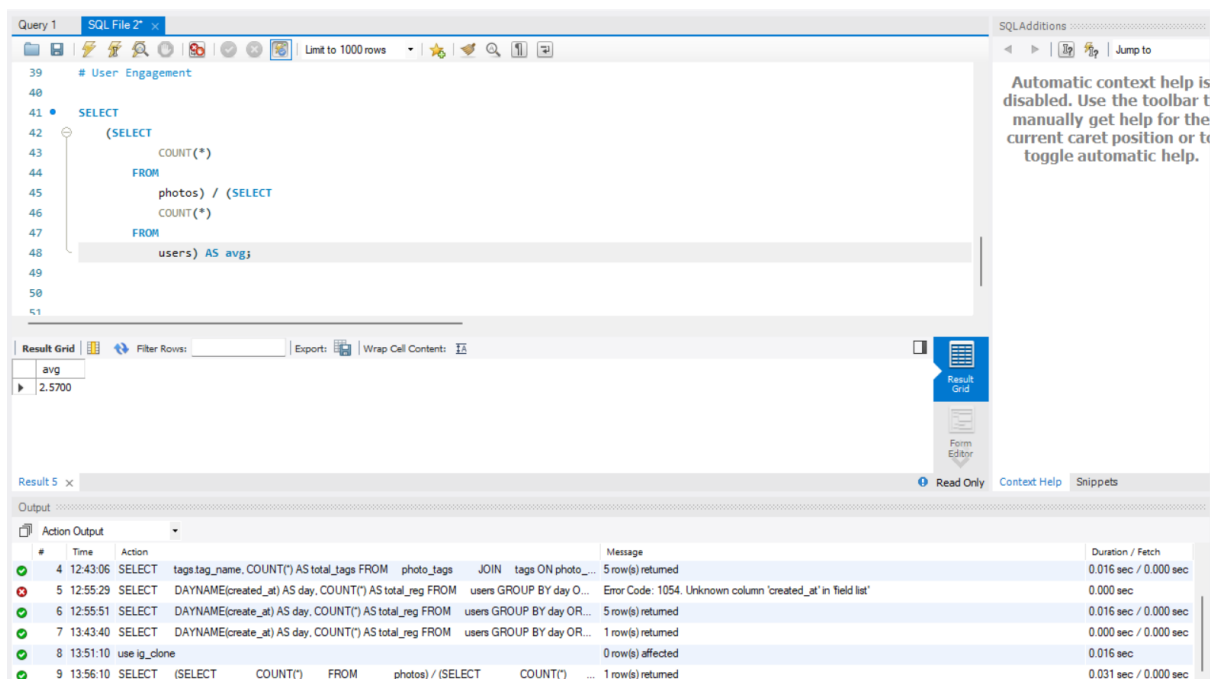
- **Objective:** Calculate the average number of posts per user and the total number of photos divided by the total number of users.
- **code:**

```

SELECT
    (SELECT
        COUNT(*)
    FROM
        photos) / (SELECT
        COUNT(*)
    FROM
        users) AS avg;

```

Result: average number of posts per user are 2.5700



The screenshot shows a SQL IDE interface with a query editor, a result grid, and an output pane. The query editor contains the following SQL code:

```

39 # User Engagement
40
41 SELECT
42     (SELECT
43         COUNT(*)
44     FROM
45         photos) / (SELECT
46         COUNT(*)
47     FROM
48         users) AS avg;
49
50
51

```

The result grid shows the following data:

avg
2.5700

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
4	12:43:06	SELECT	tags.tag_name, COUNT(*) AS total_tags FROM photo_tags JOIN tags ON photo_...	5 row(s) returned 0.016 sec / 0.000 sec
5	12:55:29	SELECT	DAYNAME(created_at) AS day, COUNT(*) AS total_reg FROM users GROUP BY day O...	Error Code: 1054. Unknown column 'created_at' in 'field list' 0.000 sec
6	12:55:51	SELECT	DAYNAME(created_at) AS day, COUNT(*) AS total_reg FROM users GROUP BY day OR...	5 row(s) returned 0.016 sec / 0.000 sec
7	13:43:40	SELECT	DAYNAME(created_at) AS day, COUNT(*) AS total_reg FROM users GROUP BY day OR...	1 row(s) returned 0.000 sec / 0.000 sec
8	13:51:10	use ig_clone		0 row(s) affected 0.016 sec
9	13:56:10	SELECT	(SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) ...	1 row(s) returned 0.031 sec / 0.000 sec

Insights: This analysis provides a baseline for user engagement and content generation trends.

2. Bots & Fake Accounts

- **Objective:** Identify users who have liked every single photo on the platform.
- **code:**

```

SELECT

    users.username, COUNT(*) AS tot_num_likes

FROM

    users

    JOIN

        likes ON users.id = likes.user_id

GROUP BY users.id

HAVING tot_num_likes = (SELECT

    COUNT(*)

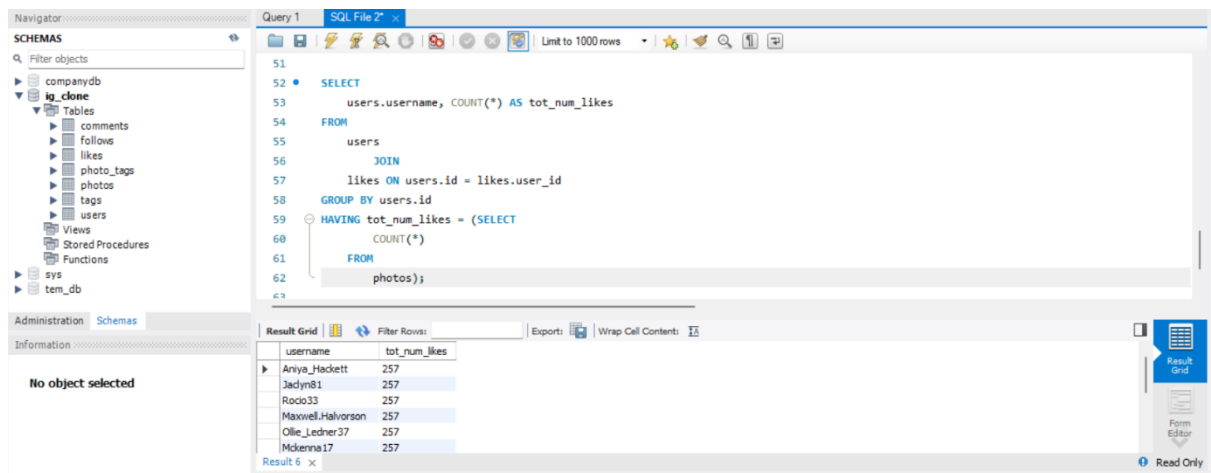
FROM

    photos);

```

Result: Identify users who have liked every single photo on the platform are as follows

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



- **Insights:** Detecting potential bots or fake accounts helps maintain platform integrity and ensures a fair user experience.

Insights:

1. Loyal User Reward:

1. By identifying the five oldest users, the platform can honor and acknowledge their contribution to the community. Rewards such as badges, special mentions, or exclusive content access can reinforce user loyalty and encourage newer users to strive for similar recognition.

2. Inactive User Engagement:

1. Inactive users represent a large pool of untapped potential. Targeted campaigns (e.g., promotional emails, exclusive content for first posts) can help convert them into active contributors, increasing overall platform engagement.

3. Contest Winner Declaration:

1. Announcing contest winners builds excitement and fosters a competitive spirit. Highlighting user achievements can also act as inspiration for other users to engage more actively with the platform.

4. Hashtag Research:

1. Hashtags play a vital role in content discovery. Providing partners and brands with data-driven hashtag recommendations ensures their campaigns resonate with a broader audience, ultimately enhancing Instagram's reputation as a marketing platform.

5. Ad Campaign Launch:

1. Insights into user registration trends allow for precise targeting of ad campaigns. This ensures that marketing efforts are optimized for visibility and reach, maximizing return on investment.

6. User Engagement Metrics:

1. Average posts per user and photos per user highlight the platform's overall engagement levels. These benchmarks help in tracking growth and ensuring that user activity scales with new registrations.

7. Bots and Fake Accounts:

1. Detecting and removing bots ensures fair competition and maintains the integrity of engagement metrics. This is crucial for investor confidence and user trust.
-

