

# **Instagram User Analytics Project Report**

## **Project Description**

This project is designed to structure, analyse, and understand user interaction and engagement data from Instagram using SQL. As a data analyst, the purpose of the project is to bring the marketing and investor teams valuable insights based on driven and quantifiable uses, ultimately facilitating informed decisions regarding business growth and improving the experience for Instagram users across the platform

## **Approach**

The project was completed by initially establishing a MySQL database—the instructions for creation of the tables, and inserting the data, were included in an SQL script advertisement. The procedure was to create the MySQL database using CREATE TABLE and INSERT INTO statements. Following the creation of the database, a set of SQL queries to answer individual and therefore specific questions posed by the marketing and investors teams was constructed, executed in MySQL Workbench, and the outputs were recorded via screenshots. The screenshots of the data were analysed and interpreted so that specific valuable insights could be derived, which are aligned in this report.

For each task, a specific SQL query was created:

- **Reward Loyal Users:** In order to obtain the oldest users, used the users table and ordered by created\_at in ascending order and limited to 5.
- **Engage Inactive Users:** In order to locate users who had not posted any photos, the users and photos table were accessed using a LEFT JOIN to isolate the NULL values in the photos.id column, implying the users had not posted photos at all.
- **Declare Contest Winners:** In order to determine the winner, I had to combine the users, photos, and likes table. The likes were counted for each respective photo\_id using GROUP BY. The result was then sorted by total\_likes which allowed me to only select the top entry.
- **Hashtags Research:** To found out which hashtags were the most popular, I needed to join tags and photo\_tags. With hashtags, I counted the usage of

the hashtag using GROUP BY tag\_name and COUNT(tag\_id), then sorted to find the top 5 hashtags.

- **Ad Campaigns Launch:** I used the DAYNAME() function on created\_at in the users tables to find the best day to launch ad campaigns. I grouped by the weekday name, and counted registrations to build an ad campaign on the day of the week that generally saw the most registrations.
- **User Engagement (Average Posts):** This metric was determined by taking the total number of all photos and dividing them by the total number of all users. Two sub-queries were needed to get the counts.
- **Bots & Fake Accounts:** Potential bot accounts were identified by joining users and likes tables, grouping by user\_id, and filtering the user only whose total likes matched the overall count of all photos on the platform using a HAVING clause.

### **Tech-Stack Used**

- **MySQL Workbench (Version 8.0.41):** This technology was selected because of the easy graphical interface, making it simple to manage a database, run queries, and visualize results simply to analyse them effectively.
- **SQL (Structure Query Language):** Was used to query and manipulate data in the relational database and allowed careful data extraction and aggregation.

### **Insights**

#### **A) Market Analysis:**

1. **Loyal User Reward:** Identify the five oldest users on Instagram.

#### **SQL Query:**

```
SELECT id, username, created_at FROM users ORDER BY created_at  
ASC LIMIT 5;
```

## Output:

The screenshot shows a SQL query editor with three queries. The first query inserts tags, the second inserts photo tags, and the third selects the five oldest users. The results of the third query are displayed in a table below the queries.

```
83
84 • INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicious'), ('beauty'), ('stunning'),
85
86
87 • INSERT 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
88
89 • SELECT id, username, created_at FROM users ORDER BY created_at ASC LIMIT 5;
90
```

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

### Insight:

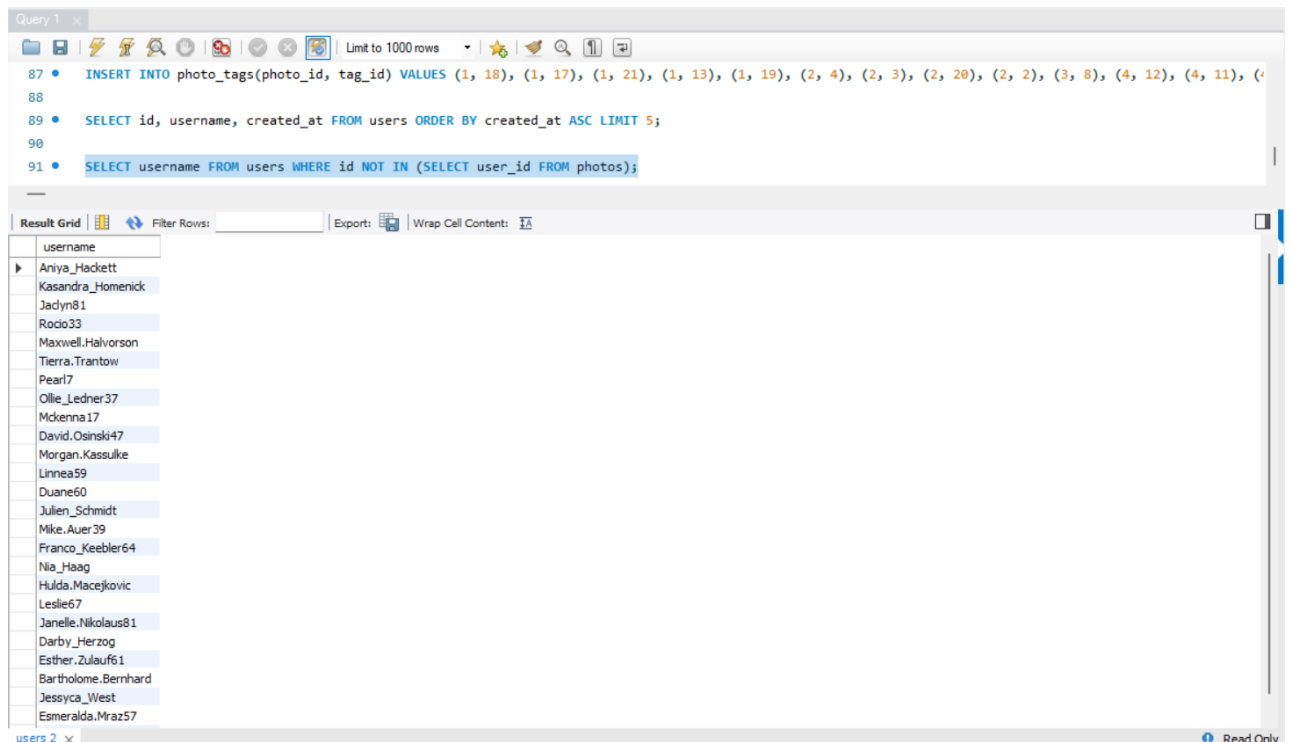
By discovering the five oldest users, the marketing team can recognize the existence of loyal users and perhaps reward them. This will allow for recognizing and appreciating these individuals and will encourage continued loyalty, especially if the reward includes exclusive offers or early access to new features.

2. Inactive User Engagement: Identify users who have never posted a single photo on Instagram.

### SQL Query:

```
SELECT username FROM users WHERE id NOT IN (SELECT user_id FROM photos);
```

Output:



Insight:

This list of inactive users is essential for the marketing team to start pursuing them again. A promotional email or notification that asks them to post could be sent to them with some luck, increasing overall user activity and content on the platform.

3. **Contest Winner Declaration:** Determine the winner of the contest (user with the most likes on a single photo).

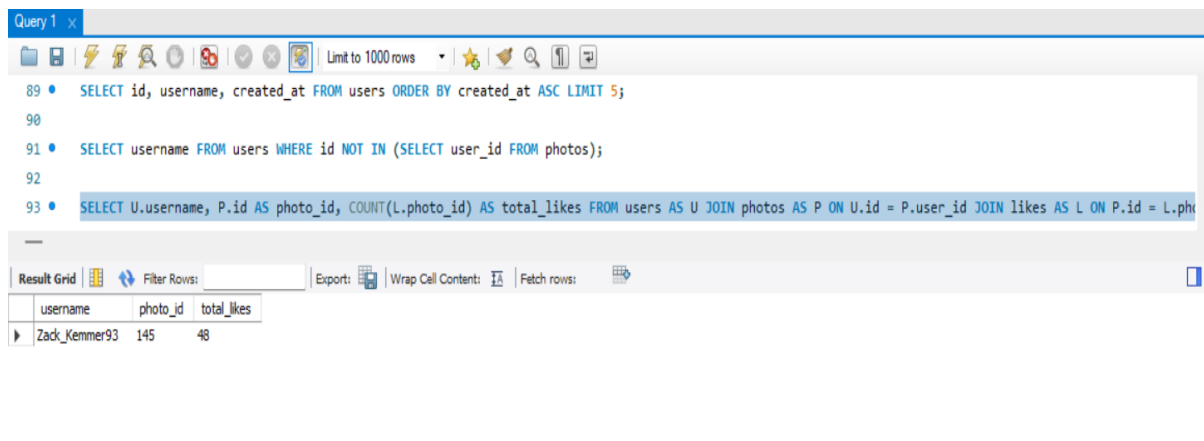
### SQL Query:

```
SELECT U.username, P.id AS photo_id, COUNT(L.photo_id) AS
total_likes FROM users AS U JOIN photos AS P ON U.id = P.user_id
JOIN likes AS L ON P.id = L.photo_id GROUP BY P.id, U.username
ORDER BY total_likes DESC LIMIT 1;
```

Insight :

It is important to be aware of the contest winner to recognize them publicly and distribute awards. This knowledge also provides an understanding of what type of content or user profile receives the greatest engagement for future content planning.

## Output:



Query 1

```
89 • SELECT id, username, created_at FROM users ORDER BY created_at ASC LIMIT 5;
90
91 • SELECT username FROM users WHERE id NOT IN (SELECT user_id FROM photos);
92
93 • SELECT U.username, P.id AS photo_id, COUNT(L.photo_id) AS total_likes FROM users AS U JOIN photos AS P ON U.id = P.user_id JOIN likes AS L ON P.id = L.pho
```

Result Grid

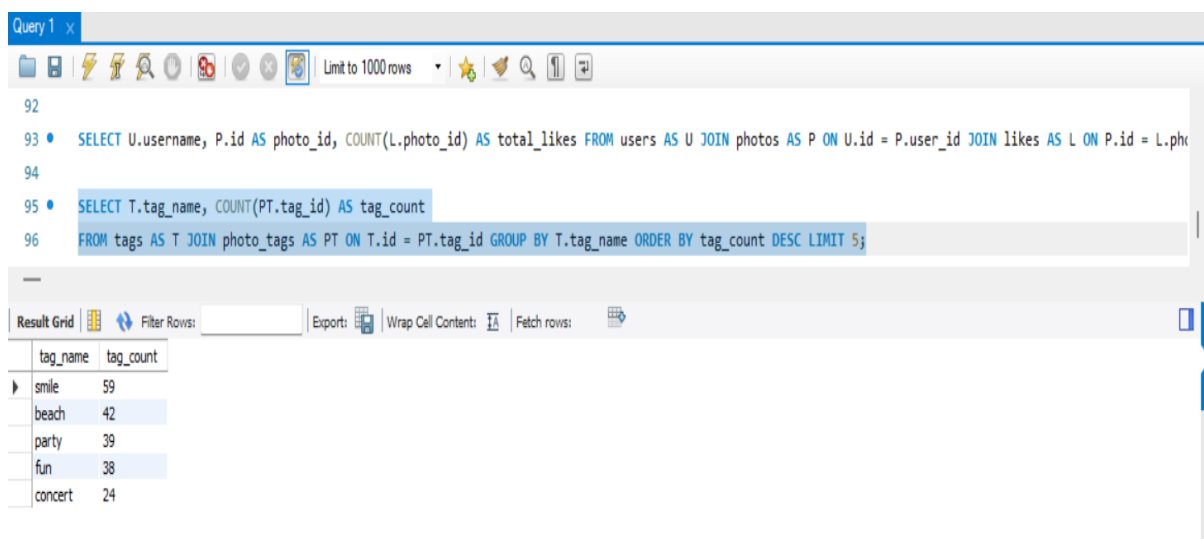
	username	photo_id	total_likes
▶	Zack_Kemmer93	145	48

4. Hashtag Research: Identify and suggest the top five most commonly used hashtags on the platform.

### SQL Query:

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

## Output:



Query 1

```
92
93 • SELECT U.username, P.id AS photo_id, COUNT(L.photo_id) AS total_likes FROM users AS U JOIN photos AS P ON U.id = P.user_id JOIN likes AS L ON P.id = L.pho
94
95 • SELECT T.tag_name, COUNT(PT.tag_id) AS tag_count
96 FROM tags AS T JOIN photo_tags AS PT ON T.id = PT.tag_id GROUP BY T.tag_name ORDER BY tag_count DESC LIMIT 5;
```

Result Grid

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

### Insights:

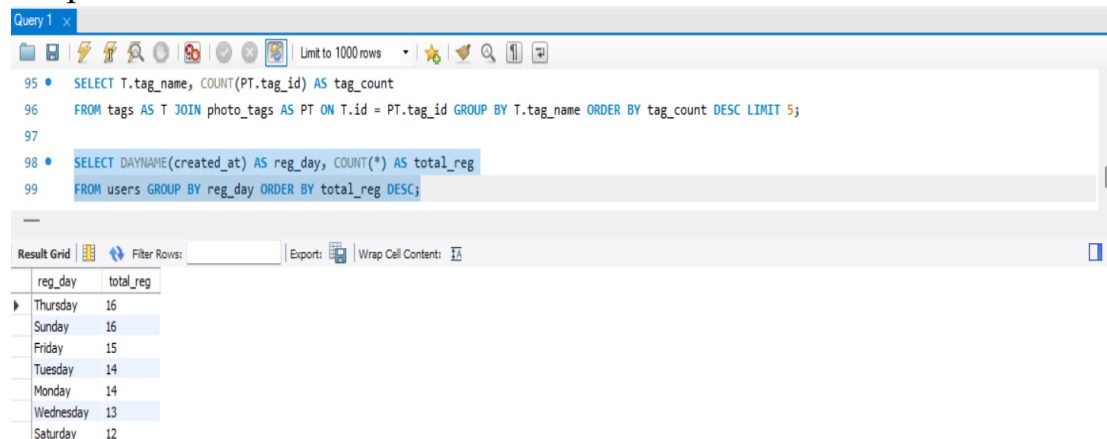
The top five most commonly used hashtags are valuable for partner brands and content creators. They indicate popular trends and topics, allowing brands to optimize their posts for maximum reach and visibility, aligning their campaigns with current user interests.

5. Ad Campaign Launch: Determine the day of the week when most users register on Instagram.

### SQL Query:

```
SELECT DAYNAME(created_at) AS reg_day, COUNT(*) AS total_reg  
FROM users GROUP BY reg_day ORDER BY total_reg DESC;
```

### Output:



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
95 • SELECT T.tag_name, COUNT(P.tag_id) AS tag_count  
96 FROM tags AS T JOIN photo_tags AS PT ON T.id = PT.tag_id GROUP BY T.tag_name ORDER BY tag_count DESC LIMIT 5;  
97  
98 • SELECT DAYNAME(created_at) AS reg_day, COUNT(*) AS total_reg  
99 FROM users GROUP BY reg_day ORDER BY total_reg DESC;
```

Below the query editor, the 'Result Grid' is displayed with the following data:

reg_day	total_reg
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

### Insights:

Being aware of the peak registration day allows the marketing team to plan shout outs strategically, by running ads on the day with the highest new users for maximum reach and benefit, potentially helping them convert more new sign ups to active use on Instagram.

## B) Investor Metrics:

1. User Engagement: Calculate the average number of posts per user on Instagram.

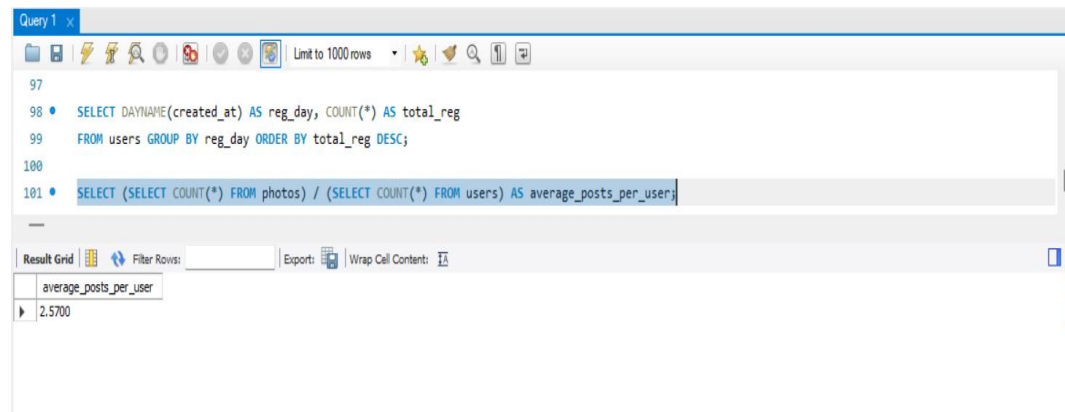
### SQL Query:

```
SELECT (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*)  
FROM users) AS average_post;
```

### Insights:

This metric provides investors with an instant view of overall user engagement. Higher averages relate to a more active and content-posting user base, which is an indicator of health for the platform, and growth potential.

Output:



```
97
98 • SELECT DAYNAME(created_at) AS reg_day, COUNT(*) AS total_reg
99   FROM users GROUP BY reg_day ORDER BY total_reg DESC;
100
101 • SELECT (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS average_posts_per_user;
```

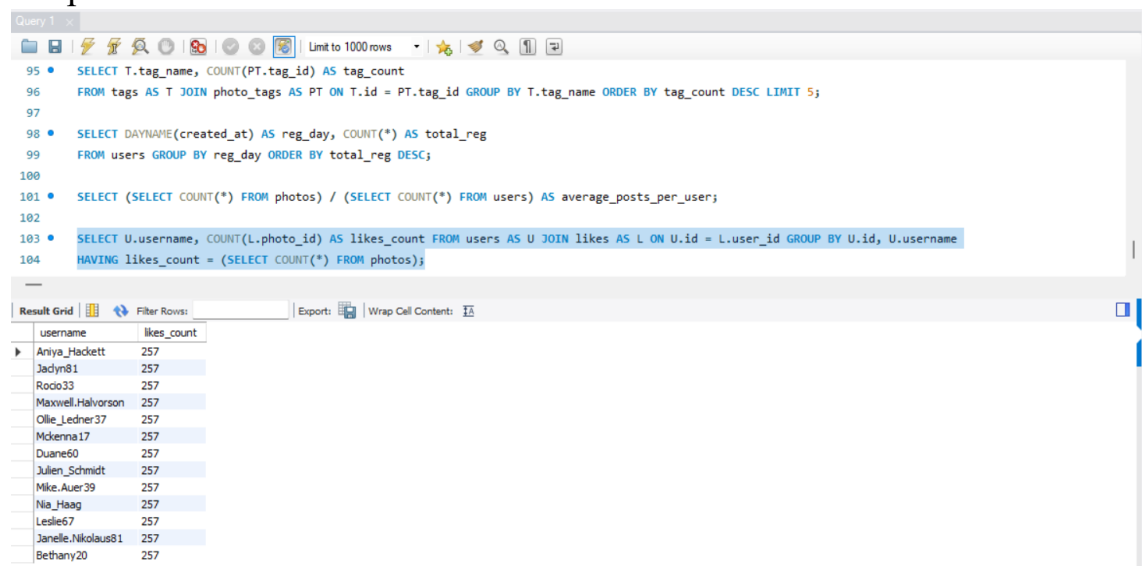
average_posts_per_user
2.5700

2. Bots & Fake Accounts: Identify users (potential bots) who have liked every single photo on the site.

### SQL Query:

```
SELECT U.username, COUNT(L.photo_id) AS likes_count FROM
users AS U JOIN likes AS L ON U.id = L.user_id GROUP BY U.id,
U.username
HAVING likes_count = (SELECT COUNT(*) FROM photos);
```

Output:



```
95 • SELECT T.tag_name, COUNT(P.tag_id) AS tag_count
96   FROM tags AS T JOIN photo_tags AS PT ON T.id = PT.tag_id GROUP BY T.tag_name ORDER BY tag_count DESC LIMIT 5;
97
98 • SELECT DAYNAME(created_at) AS reg_day, COUNT(*) AS total_reg
99   FROM users GROUP BY reg_day ORDER BY total_reg DESC;
100
101 • SELECT (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS average_posts_per_user;
102
103 • SELECT U.username, COUNT(L.photo_id) AS likes_count FROM users AS U JOIN likes AS L ON U.id = L.user_id GROUP BY U.id, U.username
104   HAVING likes_count = (SELECT COUNT(*) FROM photos);
```

username	likes_count
Aniya_Hackett	257
Jadyn81	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:

Identifying potential bot accounts is important for the integrity of the platform and user trust. This insight provides investors with knowledge

of the health of the user base and communicates a need to have strong bot deterring mechanisms in place to facilitate ongoing, non-manipulative engagement.

### **Results:**

Overall, I was able to use fundamental concepts of SQL, such as joins, aggregations, subqueries, and ordering, to generate meaningful findings from an Instagram-style dataset. I derived clear answers to vital perspectives related to the business case and confirmed the power of data analysis in answering questions for marketing and investor relations. I have improved my actual practical SQL skills and understanding of how the data can go directly into product and venture growth in a real-life situation.