

# Project Report: Instagram User Analytics



**MEGHNA HALDER**

[@meghnah22@gmail.com](mailto:@meghnah22@gmail.com)

# Project Description

## Overview:

The project involves analyzing user interactions and engagement with the Instagram app to provide valuable insights for the product team. The goal is to extract meaningful data that can influence future developments, marketing strategies, and investor relations for one of the world's most popular social media platforms.

## Purpose:

The primary purpose of this analysis is to identify key user behaviors, engagement levels, and patterns that can help the marketing team reward loyal users, engage inactive users, determine the most effective advertising times, and understand popular trends. Additionally, the project aims to provide metrics for investor insights, such as user engagement rates and identifying potential bots.

# Approach

- 01 **Data Extraction:** Utilize SQL queries to extract data from the provided Instagram database.
- 02 **Analysis:** Perform specific analyses to answer predefined questions related to marketing and investor metrics.
- 03 **Reporting:** Present the findings in a structured report with SQL queries, outputs, and explanations.

# Marketing Analysis

**Loyal User Reward:** Top 5 oldest Instagram users who have been using the platform the longest

- Steps:**
- We will use the data from the users table by selecting the username and created\_at columns.
  - Use the ORDER BY function to sort the results by the created\_at column in ascending order.
  - Use the LIMIT function to display the top 5 oldest Instagram users.

**Query:**

```
88     SELECT username, created_at
89     FROM users
90     ORDER BY created_at ASC
91     LIMIT 5;
```



**Loyal User Reward:** Top 5 oldest Instagram users who have been using the platform the longest

Output:

Result Grid			Filter Rows:	Export:	Wrap Cell Co
	username	created_at			
▶	Darby_Herzog	2016-05-06 00:14:21			
	Emilio_Bernier52	2016-05-06 13:04:30			
	Elenor88	2016-05-08 01:30:41			
	Nicole71	2016-05-09 17:30:22			
	Jordyn.Jacobson2	2016-05-14 07:56:26			
users 2 x					

**Inactive User Engagement:** identify users who have never posted a single photo on Instagram to send promotional emails.

- Steps:**
- We will choose the id and username columns from the users table.
  - We will combine the users table (u) with the photos table (p) using a left join. We will match users using u.id and photos using p.user\_id.
  - We will specify a condition to filter out users who have no corresponding entries in the photos table (p.id IS NULL). This will help us identify users who have not posted any photos.

**Query:**

```
92 • SELECT u.id, u.username
93     FROM users u
94     LEFT JOIN photos p ON u.id = p.user_id
95     WHERE p.id IS NULL;
```

**Inactive User Engagement:**

Identify users who have never posted a single photo on Instagram to send promotional emails.

**Output:**

Result Grid		
	id	username
▶	5	Aniya_Hackett
	7	Kasandra_Homenick
	14	Jadyn81
	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
	57	Julien_Schmidt
	66	Mike.Auer39
	68	Franco_Keebler64
	71	Nia_Haag

	74	Hulda.Macejkovic
	75	Leslie67
	76	Janelle.Nikolaus81
	80	Darby_Herzog
	81	Esther.Zulauf61
	83	Bartholome.Bernhard
	89	Jessyca_West
	90	Esmeralda.Mraz57
	91	Bethany20

We identified 26 inactive users who have never posted a photo on Instagram. Targeting these users with promotional emails could encourage them to engage more with the platform.

## Declaring the Contest Winner: User with the most likes on a single photo

- Steps:**
- photos, likes, and users table are joined to gather all necessary data.
  - Results are grouped by photo\_id to count likes for each photo.
  - The COUNT(likes.photo\_id) function counts the likes for each photo.
  - ORDER BY like\_count DESC sorts the photos by like count in descending order.
  - LIMIT 1 ensures only the photo with the most likes is selected.

### Query:

```
97 • SELECT
98     photos.user_id,
99     users.username,
100    photos.id AS photo_id,
101    COUNT(likes.photo_id) AS like_count
102 FROM
103     photos
104 JOIN
105     likes ON photos.id = likes.photo_id
106 JOIN
107     users ON photos.user_id = users.id
108 GROUP BY
109     likes.photo_id
110 ORDER BY
111     like_count DESC
112 LIMIT 1;
```



## Declaring the Contest Winner:

User with the most likes on a single photo

### Output:

Result Grid					Filter Rows:					Export:					Wrap Cell Content:					Fetch rows:				
	user_id	username	photo_id	like_count																				
▶	52	Zack_Kemmer93	145	48																				

The contest winner is Zack\_Kemmer93 (User ID: 52) with Photo ID: 145, receiving 48 likes, the highest in the contest. This indicates strong user engagement and popularity for Zack\_Kemmer93's photo.

## Popular Hashtag Research:

The five most commonly used hashtags on the platform for maximum reach

- Steps:**
- tags and photo\_tags table are joined to gather all necessary data.
  - The COUNT(\*) function counts the occurrences of each hashtag.
  - Results are grouped by tag\_name to aggregate usage counts.
  - ORDER BY usage\_count DESC sorts the hashtags by usage count in descending order.
  - LIMIT 5 ensures only the top five most commonly used hashtags are selected.





## Query:

```
113 • SELECT t.tag_name, COUNT(*) AS usage_count
114 FROM tags t
115 JOIN photo_tags pt ON t.id = pt.tag_id
116 GROUP BY t.tag_name
117 ORDER BY usage_count DESC
118 LIMIT 5;
```

Popular Hashtag Research:

The five most commonly used hashtags on the platform for maximum reach

Output:

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

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

The top hashtags are #smile (59 uses), #beach (42), #party (39), #fun (38), and #concert (24). These suggest a preference for positive, leisure, social, and music-related content. Using these hashtags could boost visibility and engagement for the partner brand.

## Ad Campaign Launch:

Determine the best day of the week to launch ads based on user registrations.

- Steps:**
- Used the DAYNAME function to extract the day of the week from the created\_at field in the users table.
  - Counted the number of registrations for each day of the week.
  - Grouped the results by the day of the week.
  - Ordered the results in descending order based on the number of registrations.

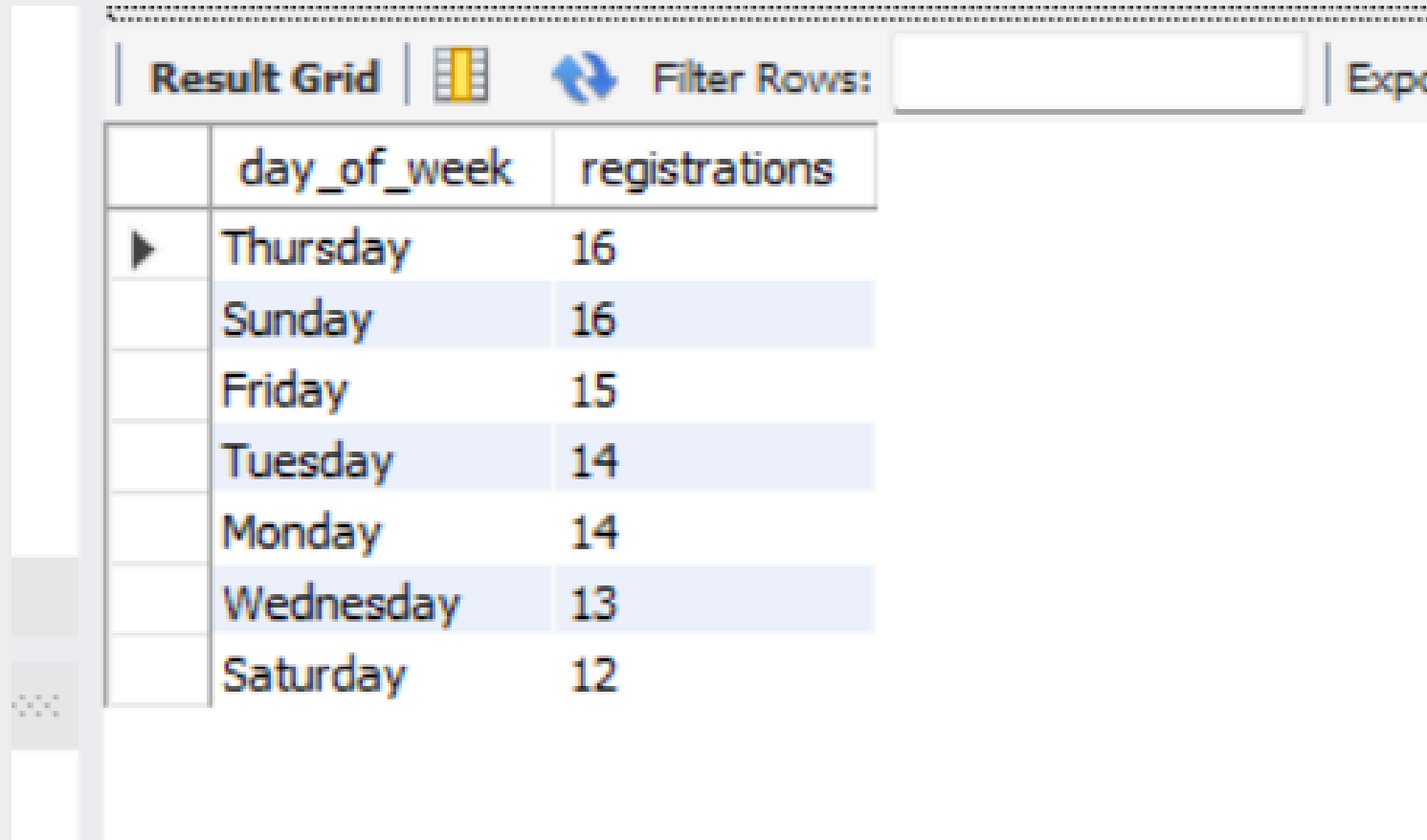
## Query:

```
120 • SELECT
121     DAYNAME(created_at) AS day_of_week,
122     COUNT(*) AS registrations
123 FROM
124     users
125 GROUP BY
126     day_of_week
127 ORDER BY
128     registrations DESC;
129
```

## Ad Campaign Launch:

Determine the best day of the week to launch ads based on user registrations.

### Output:



The screenshot shows a 'Result Grid' interface with a table of user registrations. The table has two columns: 'day\_of\_week' and 'registrations'. The data is as follows:

	day_of_week	registrations
▶	Thursday	16
	Sunday	16
	Friday	15
	Tuesday	14
	Monday	14
	Wednesday	13
	Saturday	12

Based on the data, Thursday and Sunday have the highest user registrations (16 each), followed by Friday (15), and then Monday and Tuesday (14 each). Wednesday and Saturday have the lowest registrations (13 and 12 respectively).

**Recommendation:** To achieve the best results, schedule ad campaigns on Thursday and Sunday. If additional campaign slots are needed, consider Friday as a good alternative. Avoid launching campaigns on Wednesday and Saturday due to lower user activity.



# Investor Metrics

## User Engagement:

The average number of posts per user on Instagram is calculated by dividing the total number of photos by the total number of users.

- Steps:**
- First, we count the number of posts each user has made by grouping the posts in the photos table by user\_id. This helps us determine how active each user is in terms of posting.
  - Then, we compute the average of these counts to get the overall average number of posts per user.
  - We count the total number of photos in the photos table to get the overall posting activity on the platform.
  - We count the total number of users in the users table to get the total user base.
  - We divide the total number of photos by the total number of users to get the average number of photos per user, which indicates overall user engagement.

## User Engagement:

The average number of posts per user on Instagram is calculated by dividing the total number of photos by the total number of users.

## Query:

```
130 • select
131 (select count(*) from photos)/(select count(*) from users)
132 as
133 TotalPhotos_by_TotalUsers;
134
```

## Output:

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	TotalPhotos_by_TotalUsers			
▶	2.5700			

We have 257 photos in the “photos” table and 100 IDs in the “users” table. The desired output is the ratio of photos to IDs, which can be calculated as  $257/100$  which is equal to 2.57.

User Engagement:

The average number of posts per user on Instagram is calculated by dividing the total number of photos by the total number of users.

Query:

```
135 • select user_id,count(*) as user_post_account
136 from photos
137 group by user_id
138 order by user_id;
```

Output:

	user_id	user_post_account
▶	1	5
	2	4
	3	4
	4	3
	6	5
	8	4
	9	4
	10	3
	11	5
	12	4
	13	5
	15	4
	16	4
	17	3
	18	1
	19	2

	user_id	user_post_account
	20	1
	22	1
	23	12
	26	5
	27	1
	28	4
	29	8
	30	2
	31	1
	32	4
	33	5
	35	2
	37	1
	38	2
	39	1
	40	1

	user_id	user_post_account
	42	3
	43	5
	44	4
	46	4
	47	5
	48	1
	50	3
	51	5
	52	5
	55	1
	56	1
	58	8
	59	10
	60	2
	61	1
	62	2

	user_id	user_post_account
	63	4
	64	5
	65	5
	67	3
	69	1
	70	1
	72	5
	73	1
	77	6
	78	5
	79	1
	82	2
	84	2
	85	2
	86	9
	87	4

	88	11
	92	3
	93	2
	94	1
	95	2
	96	3
	97	2
	98	1
	99	3
	100	2

## **Bots & Fake Accounts:**

Identify users (potential bots) who have liked every single photo on the site, as this is not typical behavior for a normal user.

- Steps:**
- We select the user IDs and usernames from the users table.
  - We join the users table with a subquery that finds users who have liked every photo on the site.
  - We group the records in the likes table by the user\_id to count the likes for each user.
  - We count the number of photos each user has liked using the COUNT function.
  - We compare this count to the total number of photos in the photos table to ensure that the count of liked photos for a user matches the total number of photos.
  - We use the HAVING clause in the subquery to filter out users who haven't liked every photo.
  - Finally, we return the user IDs and usernames from the main query.



## Bots & Fake Accounts:

Identify users (potential bots) who have liked every single photo on the site, as this is not typical behavior for a normal user.

### Query:

```
140 • SELECT u.id, u.username
141 FROM users u
142 JOIN (
143     SELECT l.user_id
144     FROM likes l
145     GROUP BY l.user_id
146     HAVING COUNT(l.photo_id) = (SELECT COUNT(*) FROM photos)
147 ) bot_users ON u.id = bot_users.user_id;
148
```

### Output:

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap (		
	id	username
▶	5	Aniya_Hackett
	14	Jadyn81
	21	Rocio33
	24	Maxwell.Halvorson
	36	Ollie_Ledner37
	41	Mckenna17
	54	Duane60
	57	Julien_Schmidt
	66	Mike.Auer39
	71	Nia_Haag
	75	Leslie67
	76	Janelle.Nikolaus81
	91	Bethany20

These user\_ids could be bots or fake accounts



## **Tech-Stack Used**

### **Software and Versions:**

MySQL Workbench 8.0: Chosen for its robust SQL development and management features, ideal for complex queries and large datasets.

MySQL Server 8.0: Known for reliability, performance, and compatibility with MySQL Workbench, ensuring efficient data handling.

## **Result**

### **Achievements:**

- Identified the oldest users, targeted inactive users, and declared a contest winner based on SQL analysis.
- Suggested top hashtags for partner brand visibility and optimized ad campaign launch days.
- Provided user engagement metrics and detected potential bot accounts, aiding strategic decisions.

### **Impact of Insights:**

The analysis informed strategic decisions in user retention, engagement, and marketing effectiveness, demonstrating SQL's value in deriving actionable insights from Instagram user data.