



# Instagram User Analysis

07.02.2024

Prateek Singh  
NIT Surat (2025)  
Surat, Gujrat

## Overview

I covered SQL queries for analyzing user interactions, identifying top-performing posts, and calculating engagement metrics. These discussions aimed to provide insights into user behavior, content performance, and overall engagement on Instagram.

## Goals

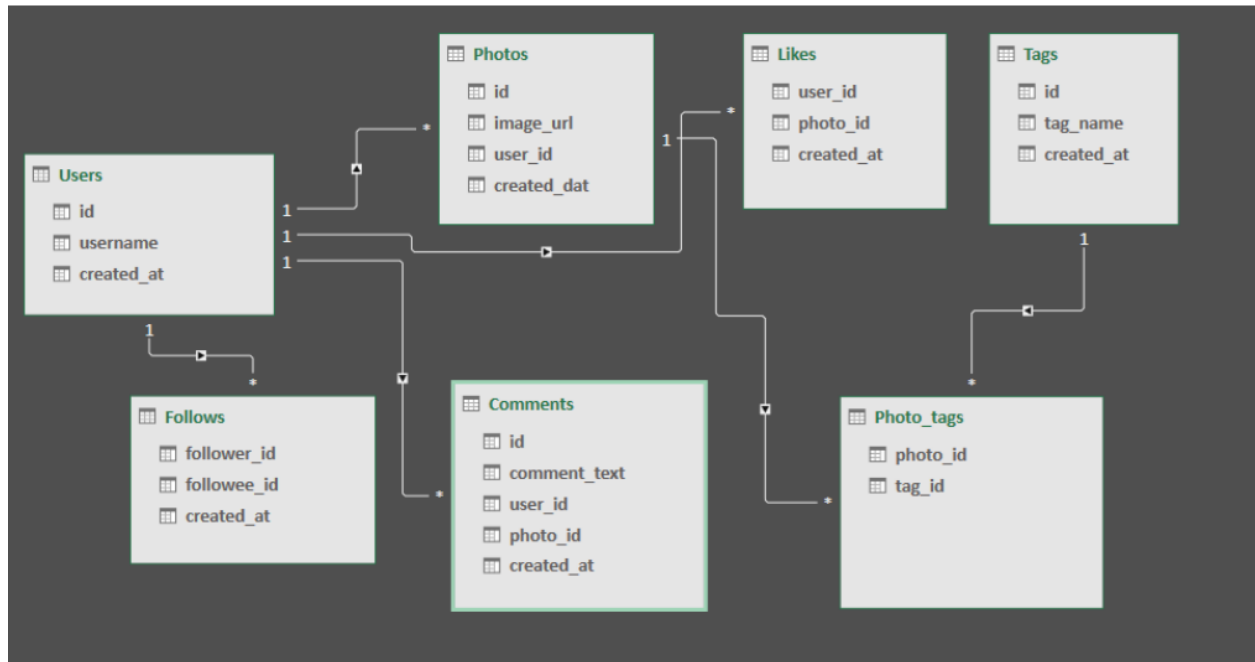
1. Marketing Analysis
2. Engagement Analysis

## Approaches

After setting up my PostgreSQL IDE(pgAdmin4), I created a database for Instagram data. Within this database, I created seven tables: COMMENTS, FOLLOWS, LIKES, PHOTO\_TAG, PHOTOS, TAGS, and USERS. Next, I imported data into these tables using the Instagram dataset obtained from Kaggle. This dataset will serve as the foundation for conducting analyses on user interactions, content performance, and overall engagement on the Instagram platform.

Here are the schemas (tables) I created in the Instagram database:

1. COMMENTS
2. FOLLOWS
3. LIKES
4. PHOTO\_TAG
5. PHOTOS
6. TAGS
7. USERS



# 1. Marketing Analysis

## 1. SEARCHING FOR MOST LOYAL USERS

Logic : Loyal users are those who have been using the service for an extended period.

### SQL Query

```

SELECT ID, USERNAME AS OLDEST_USER FROM USERS
ORDER BY CREATED_AT DESC
LIMIT 10;
  
```

Data Output			Messages	Notifications
	id [PK] integer	oldest_user character varying (50)		
1	11	Justina.Gaylord27		
2	6	Travon.Waters		
3	85	Milford_Gleichner42		
4	19	Hailee26		
5	24	Maxwell.Halvorson		
6	51	Mariano_Koch3		
7	70	Erick5		
8	55	Meggie_Doyle		
9	2	Andre_Purdy85		
10	60	Sam52		

## 2. SEARCHING FOR INACTIVE USERS

Logic :Inactive users are individuals who have not posted any photos.

### SQL Query

```
SELECT U.ID, U.USERNAME FROM USERS AS U
LEFT JOIN PHOTOS P ON
U.ID= P.user_id
WHERE image_url IS NULL;
```

Data Output			Messages		Notifications	
	id [PK] integer	username character varying (50)				
1	25	Tierra.Trantow				
2	66	Mike.Auer39				
3	89	Jessyca_West				
4	57	Julien_Schmidt				
5	34	Pearl7				
6	71	Nia_Haag				
7	83	Bartholome.Bernhard				
8	91	Bethany20				
9	21	Rocio33	18	41	Mckenna17	
10	5	Aniya_Hackett	19	54	Duane60	
11	24	Maxwell.Halvorson	20	36	Ollie_Ledner37	
12	68	Franco_Keebler64	21	53	Linnea59	
13	80	Darby_Herzog	22	49	Morgan.Kassulke	
14	76	Janelle.Nikolaus81	23	45	David.Osinski47	
15	81	Esther.Zulauf61	24	75	Leslie67	
16	90	Esmeralda.Mraz57	25	14	Jaclyn81	
17	74	Hulda.Macejkovic	26	7	Kassandra_Homenick	











## 3. MOST USED HASHTAG

Hashtags are essential in marketing for several reasons:

1. Visibility: They make content discoverable to a wider audience.
2. Brand Awareness: Unique hashtags establish brand identity and recognition.
3. Engagement: Encourage interaction and participation from users.
4. Campaign Tracking: Measure campaign performance and effectiveness.
5. Content Curation: Organize and showcase relevant user-generated content.
6. Cross-Platform Promotion: Utilize hashtags across different social media platforms for consistent messaging and broader reach.

## SQL Query

```
SELECT tag_name
FROM tags
WHERE id IN (
  SELECT tag_id
  FROM (
    SELECT tag_id, COUNT(photo_id)
    FROM photo_tags
    GROUP BY tag_id
    ORDER BY COUNT(photo_id) DESC
    LIMIT 5
  ) AS h
);
```

Data Output		Messages	Notifications
        			
	<b>tag_name</b> character varying (50) 		
1	lol		
2	fun		
3	party		
4	beach		
5	smile		

#### 4. Maximum likes on photo

### SQL Query

```
SELECT
  p.id,
  COUNT(distinct l.user_id) AS total_likes
FROM
  photos p
LEFT JOIN
  likes l ON p.id = l.photo_id
GROUP BY
  p.id
ORDER BY
  total_likes DESC
LIMIT 1;
```

Data Output			Messages	Notifications
	id [PK] integer	total_likes bigint		
1	100	8		

[photo\\_id=100](#) means [user\\_id =33](#)

```
SELECT username as winner from users
where id = 33;
```

Data Output	Messages	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	<b>winner</b> character varying (50) 🔒	
1	Yvette.Gottlieb91	

## SQL Query

```
SELECT * FROM photos
where user_id = 33;
```

Data Output	Messages	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	<b>id</b> integer 🔒	<b>image_url</b> character varying (50) 🔒
	<b>user_id</b> integer 🔒	<b>created_at</b> timestamp without time zone 🔒
1	100	https://brook.com
2	101	https://gust.net
3	102	http://bridie.name
4	103	http://barton.name
5	104	https://karina.biz

## SQL Query

```
SELECT count(follower_id) as followers FROM follows
where followee_id = '33';
```

Data Output	Messages	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	<b>followers</b> bigint 🔒	
1	10	

## 2. Engagement Analysis

Logic : We will select the user\_id from the likes table where the number of likes is maximum, and select the user\_id from the comments table where the user has made the maximum number of comments. Then, calculate the sum of these two values. The user\_id with the maximum sum of likes and comments represents the customer with the highest engagement.

### 1. Like analysis

#### SQL Query

```
SELECT user_id, COUNT(*) AS total_likes_given
FROM likes
GROUP BY user_id
ORDER BY total_likes_given DESC;
```

	user_id integer	total_likes_given bigint
1	5	257
2	2	94
3	4	93
4	11	89
5	10	87
6	9	85
7	6	82
8	8	79
9	3	79
10	12	55

### 2. Comment analysis

#### SQL Query

```
SELECT user_id, COUNT(*) AS total_comments_made
FROM comments
GROUP BY user_id
```



```
ORDER BY total_comments_made DESC  
limit 10;
```

	user_id integer	total_comments_made bigint
1	36	35
2	24	35
3	21	35
4	5	35
5	14	35
6	75	34
7	71	34
8	41	34
9	66	34
10	54	34

### 3. Following analysis

#### SQL Query

```
SELECT follower_id AS user_id, COUNT(*) AS total_followers  
FROM follows  
GROUP BY follower_id  
ORDER BY total_followers DESC;
```

	user_id integer	total_followers bigint
1	6	99
2	3	99
3	11	99
4	8	99
5	12	99
6	10	99
7	9	99
8	5	99
9	4	99
10	2	99
11	13	10

#### 4. Follower analysis

##### SQL Query

```
SELECT followee_id AS user_id, COUNT(*) AS total_followers
FROM follows
GROUP BY followee_id
ORDER BY total_followers DESC;
```

	user_id integer 🔒	total_followers bigint 🔒
1	1	11
2	7	11
3	23	10
4	56	10
5	91	10
6	58	10
7	8	10
8	87	10
9	74	10
10	54	10
11	29	10
12	71	10
13	68	10
14	4	10
15	34	10
16	51	10
17	96	10

### Overall engagement score









#### SQL Query

```
SELECT
    likes_data.user_id,
    (total_likes_given + total_comments_made) AS engagement_score
FROM (
    SELECT user_id, COUNT(*) AS total_likes_given
    FROM likes
    GROUP BY user_id
) AS likes_data
JOIN (
```

```

SELECT user_id, COUNT(*) AS total_comments_made
FROM comments
GROUP BY user_id
) AS comments_data ON likes_data.user_id = comments_data.user_id
WHERE likes_data.user_id != 5 -- Exclude user_id = 5
ORDER BY engagement_score DESC;

```

Data Output	Messages	Notifications
		
		
		
user_id integer	engagement_score bigint	
1	2	101
2	4	99
3	11	95
4	10	93
5	9	92
6	6	88
7	8	88
8	3	87
9	12	63









*Important to note :I excluded user\_id=5 since it's associated with an inactive user, possibly indicating it as a bot.*

## SQL Query

```

SELECT * FROM users
WHERE id = 2;

```

Data Output	Messages	Notifications
		
		
		
id [PK] integer	username character varying (50)	created_at timestamp without time zone
1	2 Andre_Purdy85	2017-04-02 17:11:21

"Thus, user\_id= 2 ie **Andre\_Purdy85** exhibits the highest level of engagement."

## **Result**

This project make me delve to learned

1. Data Modeling Proficiency: Recognizing the interrelationships among data schemas facilitates insightful analysis and informed conclusions. This foundational step lays the groundwork for navigating complex datasets effectively.

2. Alias Utilization: Employing meaningful aliases enhances code clarity and aids in understanding query structures. Strategic aliasing serves as a mnemonic device, aiding in recall and improving overall code readability.

3. Command Proficiency: Hands-on experience with various SQL commands enhances problem-solving capabilities and boosts confidence in tackling diverse data challenges. Practical application fosters a deeper understanding of command functionalities and their practical implications.

4. Real-World Project Engagement: Participation in sample projects mirrors authentic data scenarios, providing invaluable experience in addressing real-world issues. This exposure fosters the development of practical problem-solving skills and the ability to identify meaningful patterns within datasets.

5. Insightful Analysis: By leveraging data modeling skills, alias usage, command proficiency, and real-world project experience, individuals gain the capability to extract meaningful insights. This holistic skill set empowers effective data analysis and informed decision-making processes.