

Instagram User Analytics

- **Project Description:** The goal of this project was to analyze Instagram's user and engagement data in order to derive meaningful insights about marketing strategies and investor-focused metrics. The tasks were executed systematically using SQL queries in order to ensure data accuracy and generate actionable insights.

- **Approach:**

A) Marketing Analysis:

- **Loyal User Reward:** The marketing team aims to reward the most loyal users, i.e., those who have been using the platform for the longest time. To achieve this, I identified the five oldest users on Instagram by analyzing the **created_at** field in the provided **Users** table. This was done using the **ORDER BY** clause to sort the users based on their registration date and time in ascending order, ensuring that the longest-serving users were accurately identified.
- **Inactive User Engagement:** The team aims to encourage inactive users to start posting by sending them promotional emails. To achieve this, I identified users who have never posted a single photo on Instagram. This was accomplished by analyzing the **Users** table and the **Photos** table. Using a **LEFT JOIN** between the two tables and filtering for **user_id** values that are **NULL** in the **Photos** table, inactive users were accurately identified, ensuring their precise selection for the promotional campaign.
- **Contest Winner Declaration:** The team has organized a contest where the user with the most likes on a single photo win. To determine the winner, I identified the photo with the highest number of likes by analyzing the **Likes** table and calculating the like count for each photo. Using this data, I joined the **Likes** table with the **Photos** table, and then retrieved the user details from the **Users** table by further **joining** it with the **Photos** table. This ensured the accurate identification of the winner. The user details, including their **photo_id**, **username**, and **number of likes**, were then provided to the team for the announcement.

- **Hashtag Research:** A partner brand wants to know the most popular hashtags to use in their posts to reach the most people. To achieve this, I analyzed the **Photo_tags** and **Tags** tables to identify the top five most commonly used hashtags on the platform. By counting the occurrences of each **tag_id**, grouping by **tag_name**, and ordering them in descending order, I identified the most frequently used hashtags. This provided the brand with the most effective hashtags to improve post visibility and reach a larger audience.
- **Ad Campaign Launch:** The team wants to know the best day of the week to launch ads. To determine this, I analyzed the **Users** table and identified the day of the week when most users registered on Instagram. Using SQL functions like **DAYNAME** to extract the day of the week and **COUNT** to measure the number of registrations, I found that **Thursdays** had the highest registration activity. This insight makes Thursday the ideal day to schedule ad campaigns for maximum reach and engagement.

B) Investor Metrics

- **User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts. To provide this insight, I calculated the **average number of posts per user** on Instagram by dividing the total number of photos by the total number of users. This metric shows how active users are on the platform and whether engagement is increasing or decreasing.
- **Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts. To detect potential bots, I identified users who have liked every single photo on the platform, as this is unusual behavior for a typical user. By comparing the total number of photos and likes per user, I identified users whose activity matched this pattern.

- **Tech-Stack Used**

MySQL Workbench (Version 8.0): Used for writing and executing SQL queries due to its user-friendly interface and ability to visualize results.

- **Insights**

- A) Marketing Analysis**

- **Loyal User Reward:** Identified the five oldest users based on registration dates, highlighting loyal users who can be rewarded to strengthen engagement and retention.
 - **Inactive User Engagement:** Highlighted inactive users, providing an opportunity to re-engage them with personalized campaigns.
 - **Contest Winner Declaration:** Determine the winner of the contest based on the highest photo likes, showcasing their popularity and engagement.
 - **Hashtag Research:** Discovered the top five most-used hashtags, providing valuable input for optimizing post visibility.
 - **Ad Campaign Launch:** Thursdays were identified as the peak registration day, making it an ideal time to schedule ad campaigns.

- B) Investor Metrics**

- **User Engagement:** Measured content generation and user activity trends, showing healthy engagement rates and platform activity.
 - **Bots & Fake Accounts:** Identified suspicious accounts with bot-like behavior, ensuring better platform integrity and reliability for investors.

- **Result:**

A) Marketing Analysis:

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

The screenshot shows a SQL IDE window titled "IG_Data" with a tab "Instagram_User_Analytics* x". The query editor contains the following SQL code:

```
1 • USE ig_clone;
2
3 -- A) Marketing Analysis:
4
5 /* 1. Loyal User Reward:
6 Identify the five oldest users on Instagram from the provided database.*/
7
8 • SELECT * FROM Users ORDER BY created_at LIMIT 5;
9
10 /* 2. Inactive User Engagement:
11 Identify users who have never posted a single photo on Instagram*/
12 /* 3. Contest Winner Declaration:
```

Below the query editor is the "Result Grid" showing the results of the query. The grid has columns "id", "username", and "created_at". The results are as follows:

	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

The five oldest users on Instagram are **User IDs 80, 67, 63, 95, and 38.**

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

IG_Data | Instagram_User_Analytics* x

Limit to 1000 rows

```
10 /* 2. Inactive User Engagement:
11 Identify users who have never posted a single photo on Instagram*/
12
13 • SELECT
14     u.id, u.username
15 FROM
16     Users u
17     LEFT JOIN
18     Photos p ON u.id = p.user_id
19 WHERE
20     p.user_id IS NULL;
```

Result Grid

	id	username
5	Aniya_Hackett	
7	Kassandra_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	

Result Grid

	id	username
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	

There is total **26 users** who have never posted a single photo on Instagram.

```
22 • SELECT
23     count(*)
24 FROM
25     Users u
26     LEFT JOIN
27     Photos p ON u.id = p.user_id
28 WHERE
29     p.user_id IS NULL;
30
```

Result Grid

	count(*)
26	

- 3. Contest Winner Declaration:** Determine the winner of the contest and provide their details to the team (The user with the most likes on a single photo wins).

IG_Data Instagram_User_Analytics* x

Limit to 1000 rows

```
33 /* 3. Contest Winner Declaration:
34 Determine the winner of the contest and provide
35 (The User with the most likes on a single photo
36
37 • SELECT
38     likes.photo_id,
39     users.username,
40     COUNT(likes.user_id) AS no_of_likes
41 FROM
42     likes
43     INNER JOIN
44     photos ON likes.photo_id = photos.id
45     INNER JOIN
46     users ON photos.user_id = users.id
47 GROUP BY likes.photo_id , users.username
48 ORDER BY no_of_likes DESC;
49
```

Result Grid

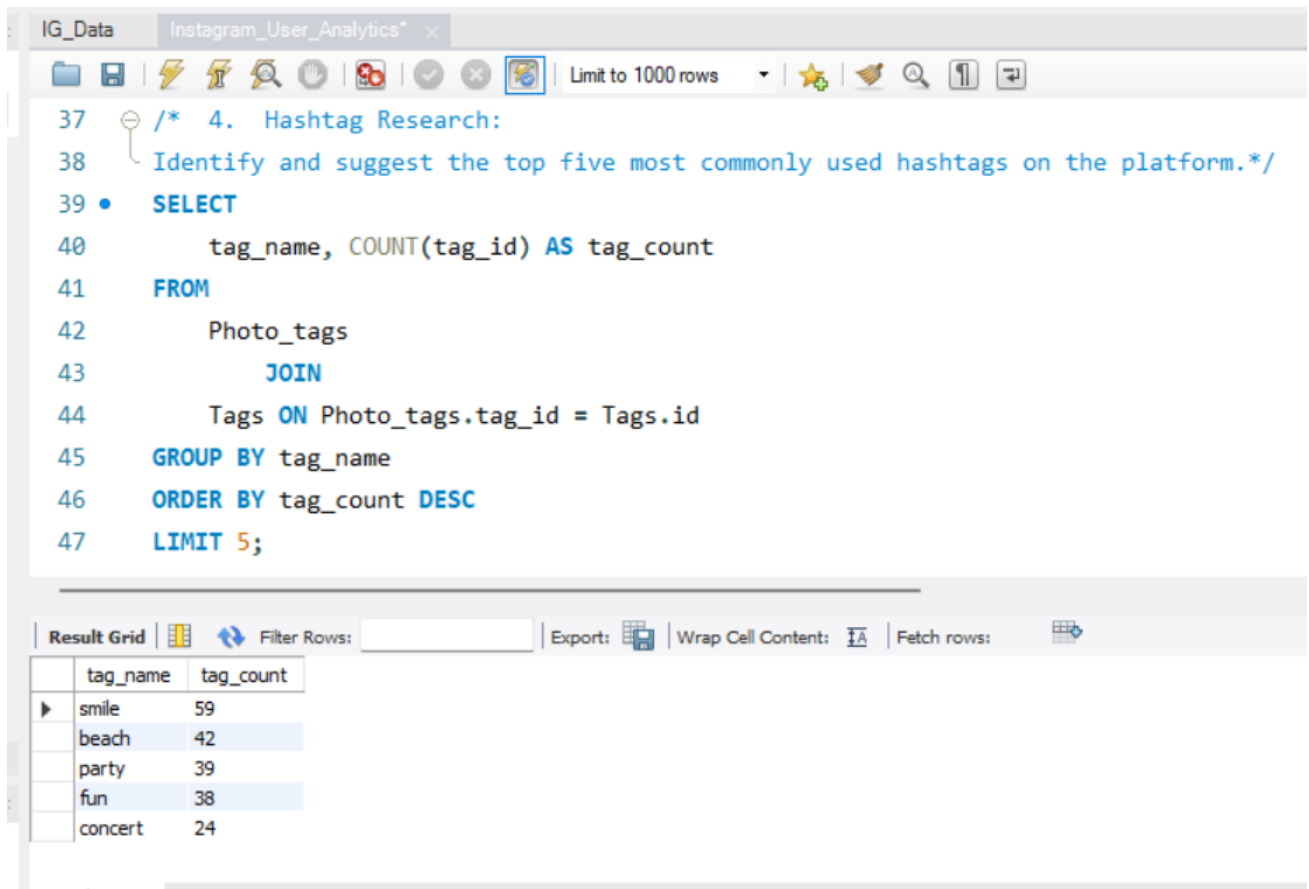
photo_id	username	no_of_likes
145	Zack_Kemmer93	48
127	Malinda_Streich	43
182	Adelle96	43
123	Seth46	42
30	Presley_McClure	41
52	Annalise.McKenzie16	41
61	Delpha.Kihn	41
147	Meggie_Doyle	41

Result Grid

Filter Rows:

	photo_id	username	no_of_likes
▶	145	Zack_Kemmer93	48
	127	Malinda_Streich	43
	182	Adelle96	43
	123	Seth46	42
	30	Presley_McClure	41
	52	Annalise.McKenzie16	41
	61	Delpha.Kihn	41
	147	Meggie_Doyle	41
	174	Elenor88	41
	192	Kathryn80	41
	256	Javonte83	41
	13	Harley_Lind18	40
	97	Irwin.Larson	40
	153	Aurelie71	40
	161	Cesar93	40
	244	Damon35	40
	44	Alexandro35	39
	62	Kenneth64	39
	66	Eveline95	39
	100	Yvette.Gottlieb91	39
	107	Yazmin_Mills95	39
	110	Kelsi26	39
	118	Janet.Armstrong	39
	119	Janet.Armstrong	39
	144	Zack_Kemmer93	39

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



The screenshot shows a SQL query editor with a query to find the top 5 most common hashtags. The query is as follows:

```
37  /* 4. Hashtag Research:
38  Identify and suggest the top five most commonly used hashtags on the platform.*/
39  • SELECT
40      tag_name, COUNT(tag_id) AS tag_count
41  FROM
42      Photo_tags
43      JOIN
44      Tags ON Photo_tags.tag_id = Tags.id
45  GROUP BY tag_name
46  ORDER BY tag_count DESC
47  LIMIT 5;
```

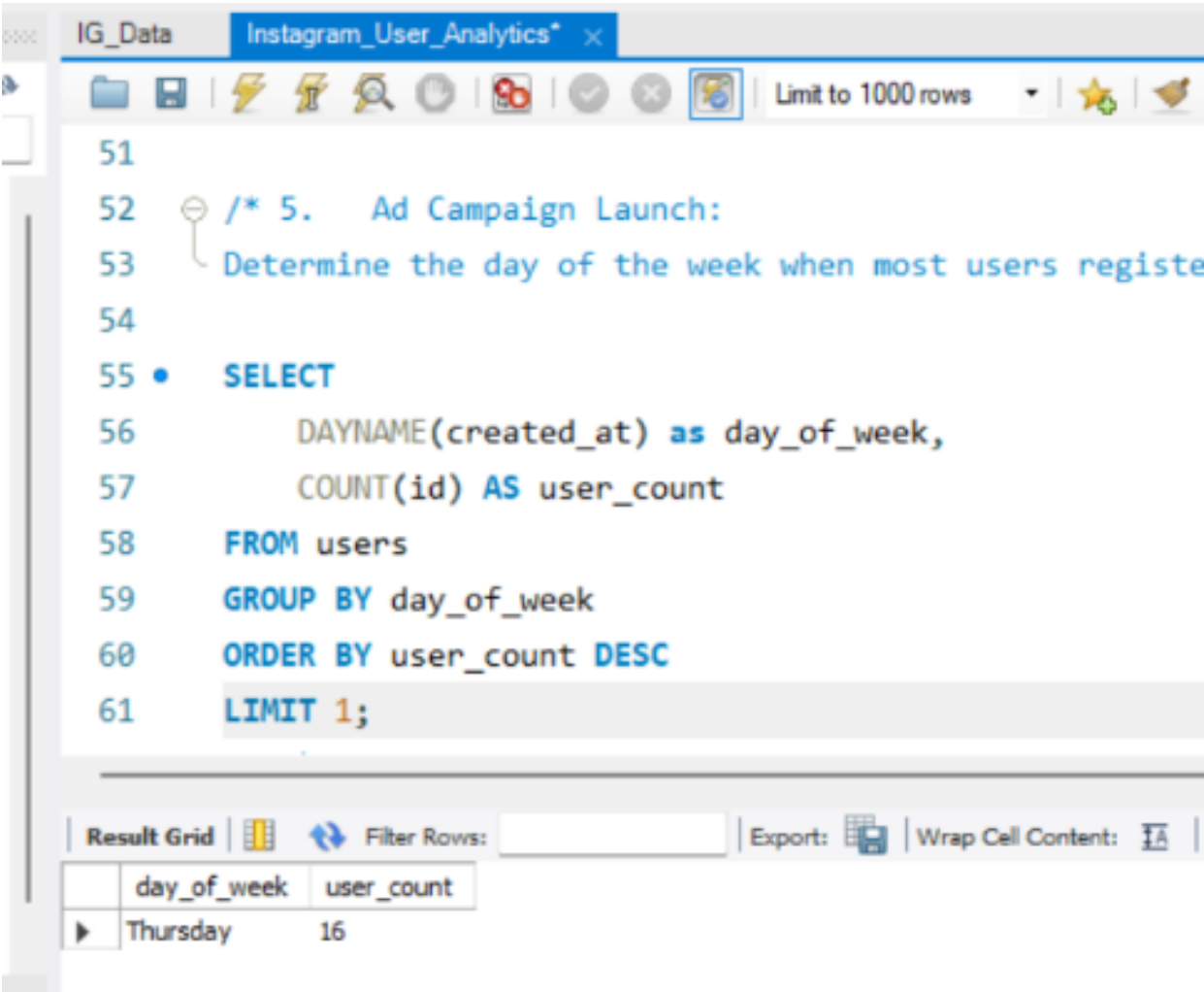
Below the query editor, the result grid is displayed, showing the top 5 most common hashtags and their counts:

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

Smile, Beach, Party, Fun, Concert are the top five most commonly used hashtags on the platform.

5. Ad Campaign Launch:

Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.



The screenshot shows a SQL query editor with a tab labeled "Instagram_User_Analytics*". The query is as follows:

```
51
52 /* 5. Ad Campaign Launch:
53 Determine the day of the week when most users registe
54
55 • SELECT
56     DAYNAME(created_at) as day_of_week,
57     COUNT(id) AS user_count
58 FROM users
59 GROUP BY day_of_week
60 ORDER BY user_count DESC
61 LIMIT 1;
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query:

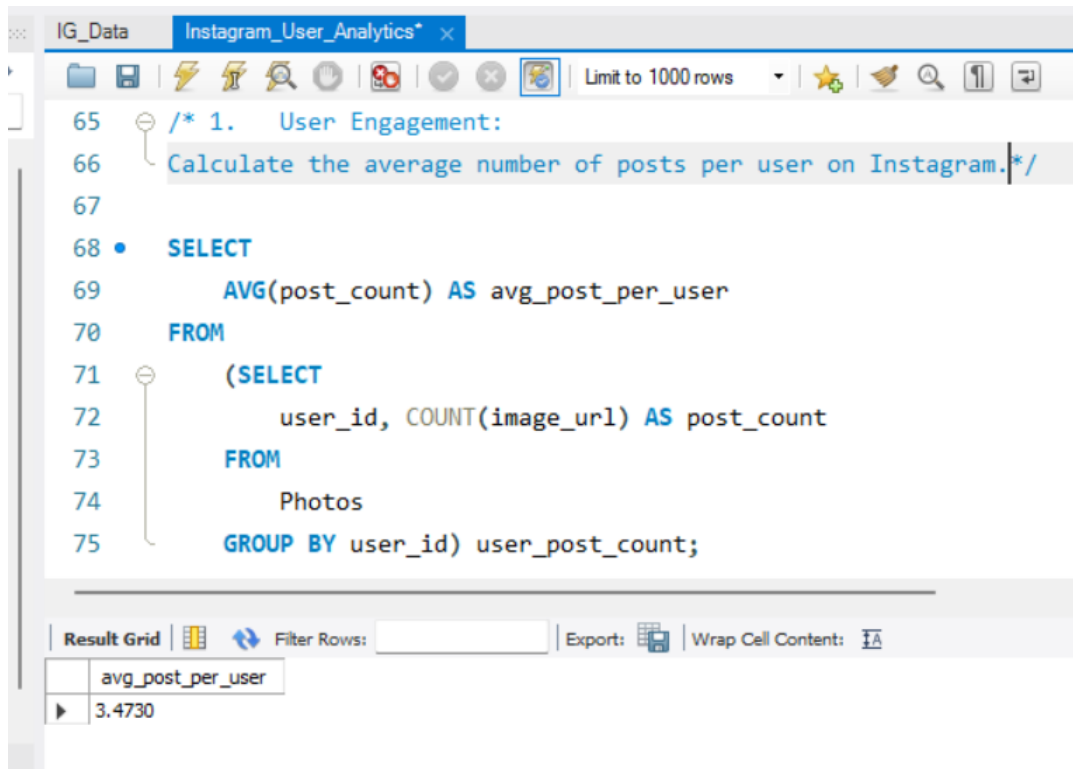
day_of_week	user_count
Thursday	16

Most of the users register on Instagram **on Thursday**

B) Investor Metrics:

1. User Engagement:

Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

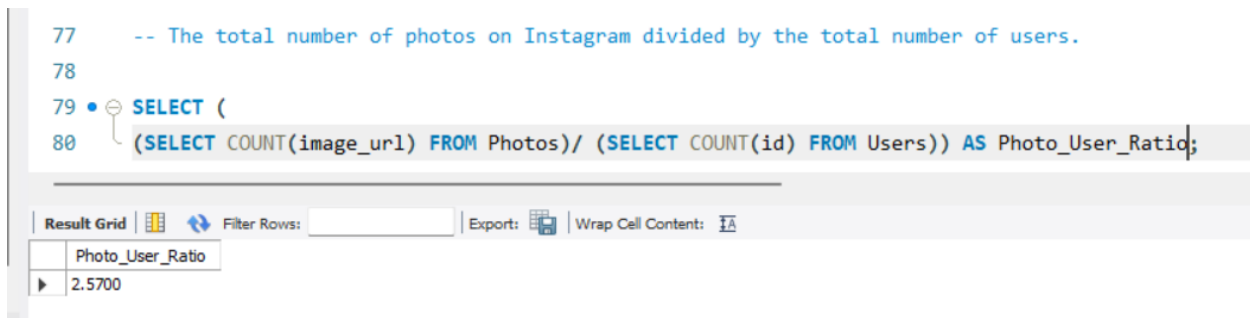


```
65 /* 1. User Engagement:
66 Calculate the average number of posts per user on Instagram.*/
67
68 • SELECT
69     AVG(post_count) AS avg_post_per_user
70 FROM
71     (SELECT
72         user_id, COUNT(image_url) AS post_count
73     FROM
74         Photos
75     GROUP BY user_id) user_post_count;
```

Result Grid

avg_post_per_user
3.4730

The average number of posts per user on Instagram is **3.4730**



```
77 -- The total number of photos on Instagram divided by the total number of users.
78
79 • SELECT (
80     (SELECT COUNT(image_url) FROM Photos) / (SELECT COUNT(id) FROM Users)) AS Photo_User_Ratio;
```

Result Grid

Photo_User_Ratio
2.5700

The Photo-User Ratio is **2.5700**

2. Bots & Fake Accounts:

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

```
81
82 /* 2. Bots & Fake Accounts:
83 Identify users (potential bots) who have liked every single photo on the site, as this is not typically
84 possible for a normal user.*/
85
86 • SELECT user_id
87 FROM
88     Likes
89 GROUP BY user_id
90 HAVING COUNT(photo_id) = (SELECT COUNT(id) FROM Photos);
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	user_id
▶	5
	14
	21
	24
	36
	41
	54
	57
	66
	71
	75
	76
	91

Likes 52 ×

- Drive Link

Dataset:

https://drive.google.com/file/d/1_g5WfqGCzsedJzpgRIoJNdZSJCdLRPX/view?usp=sharing

SQL File:

https://drive.google.com/file/d/1EsCQ_cmFeDSnkPpmlbVZErqwgVh0DJv6/view?usp=sharing