

INSTAGRAM USER ANALYTICS

Description

The project involves analysing user interactions and engagement with the Instagram app to provide valuable insights that can help the business grow. By utilizing SQL to query a database containing Instagram user data, aim to understand how user interact with the platform, identify trends, and generate actionable insights to improve engagement strategies. The goal of this project is to use SQL to extract meaningful insights from the data.

The Problem

A) Marketing Analysis:

- Loyal User Reward: The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

Your Task: Identify the five oldest users on Instagram from the provided database.

- Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.

Your Task: Identify users who have never posted a single photo on Instagram.

- Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.

Your Task: Determine the winner of the contest and provide their details to the team.

- Hashtag Research: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

- Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

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

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.

Your Task: 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.

- Bots & Fake Accounts: Investors want to know if the platform is crowded with fake and dummy accounts.

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

Design

Steps to Load Data into the Database

- Create a Database.
 - *Use the CREATE DATABASE command in MySQL to set up a new database.*
- Create Tables and Define Columns.
 - *Use the CREATE TABLE command to structure tables with appropriate column names and data types.*
- Insert Data into Tables.
 - *Use the INSERT INTO command to populate tables with user data, posts, likes, and other relevant information.*
- Query the Data.
 - *Use the SELECT command to retrieve and analyze specific insights from the database.*

Software used for querying the results

- MySQL Workbench 8.0 CE

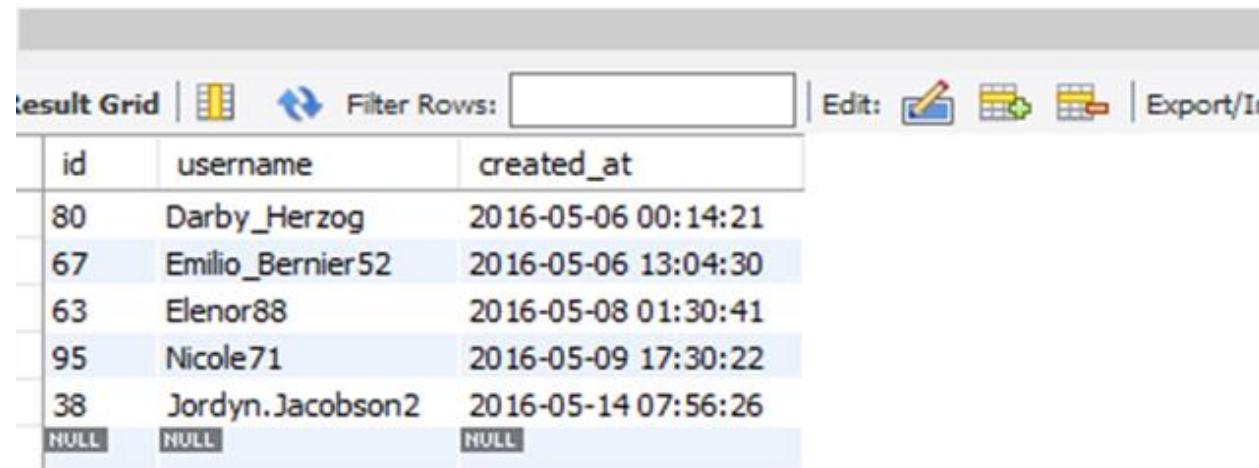
Findings-1

To find the most loyal i.e. the top 5 oldest users of Instagram:

- We will use the data from the **users** table by executing a **SELECT** query to fetch the required details.
- Then, using the **ORDER BY** function, we will sort the output by the **created_at** column in ascending order.
- Then using the **limit** function, the output will be displayed for top 5 oldest Instagram users.

Output

```
4 •      select * from users order by created_at asc limit 5;
```



The screenshot shows the MySQL Workbench interface with the results of a query execution. The query is displayed in the SQL editor at the top. Below it, the 'Result Grid' tab is active, showing a table with three columns: 'id', 'username', and 'created_at'. The results are listed 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

Findings-2

To find the most inactive users, i.e., users who have never posted a single photo on Instagram:

- We will first select the **username** column from the **users** table to identify inactive users.
- Then, we will **LEFT JOIN** the **photos** table with the **users** table on **users.id = photos.user_id** since both columns share common values. This will help identify users who have never posted a photo.
- Then, we will find rows from the **users** table where the **image_url** is **NULL**, indicating users who have never posted a photo.

Output

```
8   # Identify users who have never posted a single photo on Instagram.  
9 * select username from users  
10  left join photos on users.id=photos.user_id where image_url is null;  
11
```

username
Aniya_Hackett
Kassandra_Homenick
Jadyn81
Rocio33
Maxwell_Halvorson
Tierra.Trantow

Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike_Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67

Result 65 X

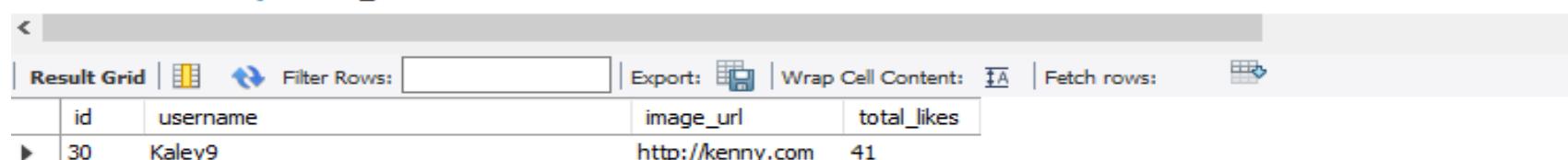
Findings-3

To find the **username**, **photo_id**, **image_url**, and **total_number_of_likes** for each image, we will join the **users**, **photos**, and **likes** tables.

- First we will select the **users.username**, **users.id**, **photos.image_url** and **count(*) as total_likes**.
- Then, we will **INNER JOIN** the three tables: **photos**, **likes**, and **users** using the conditions **photos.id = likes.photo_id** and **users.id = likes.photo_id**.
- Then, by using the **GROUP BY** function, we will group the output based on **photos.id**.
- Then, using the **ORDER BY** function, we will sort the data based on the **total_likes** in descending order to display the most liked photos first.
- Then, to find the most liked photo, we will use the **LIMIT** function to display only the top liked photo's information.

Output

```
15 •   select users.id, users.username,photos.image_url,count(*) as total_likes from likes
16     inner join photos on photos.id=likes.photo_id
17     inner join users on users.id=likes.photo_id
18     group by photo_id
19     order by total_likes desc limit 1;
```



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the SQL code provided above. The result grid displays a single row of data:

	id	username	image_url	total_likes
▶	30	Kaley9	http://kenny.com	41

Findings-4

To find the top 5 most commonly used hashtags on Instagram:

- We need to select the **tag_name** column from the **tag** table and use the **COUNT(*)** function to count the number of times each tag has been used individually.
- Then, we need to **INNER JOIN** the **tags** table and **photo_tags** table on **photo_tags.tag_id = tags.id** because they share the common column **tag_id**.
- Then, using the **GROUP BY** function, we need to group the desired output based on **tag_name**
- Then, using the **ORDER BY** function, we need to sort the output based on the **total_num_used**, in descending order.
- Then, to find the top 5 most used tag names, we will use the **LIMIT 5** function to display only the five most frequently used tags.

Output

```
24 •      select tag_name, count(*) as total_num_used from tags
25      inner join photo_tags on photo_tags.tag_id=tags.id
26      group by tag_name
27      order by total_num_used desc limit 5;
```

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

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

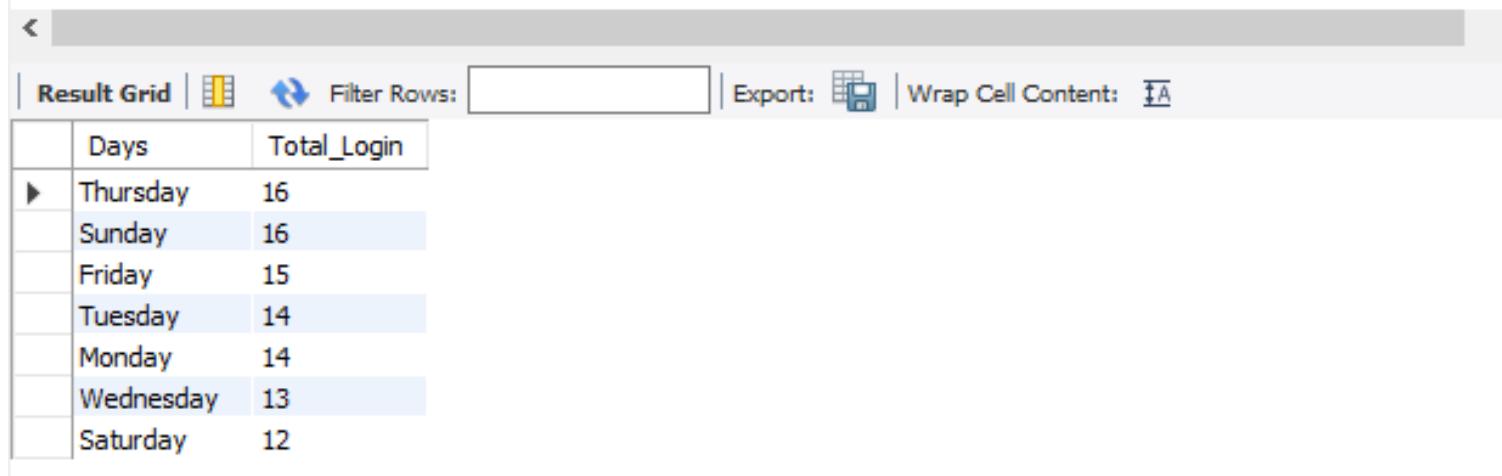
Findings-5

To find the day of week on which most users register on Instagram:

- First we define the columns of the desired output table using **SELECT dayname(created_at) as Days** and **count(*) as Total_logins** from the **users** table.
- Then, using the **GROUP BY** function, we group the output table based on **Days**.
- Then, using the **ORDER BY** function, we sort the output table based on **Total_logins** in descending order to display the days with the highest login first.

Output

```
35 •   select dayname(created_at) as Days, count(*) as Total_Login from users  
36     group by Days order by Total_Login desc ;
```



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
35 •   select dayname(created_at) as Days, count(*) as Total_Login from users  
36     group by Days order by Total_Login desc ;
```

The result grid displays the following data:

	Days	Total_Login
▶	Thursday	16
	Sunday	16
	Friday	15
	Tuesday	14
	Monday	14
	Wednesday	13
	Saturday	12

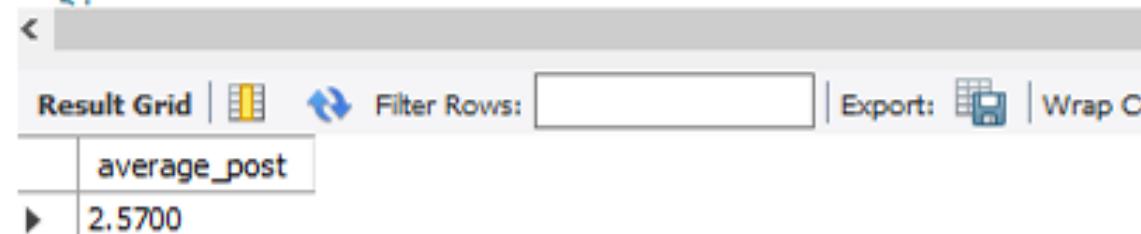
Findings-6

To find the average number of posts per user on Instagram:

- First, we need to find the total number of photos (posts) present in the photos table by counting the **photos.id** column.
- Similarly, we need to find the total number of users present in the users table by counting the **users.id** column.
- Next, we need to divide the total number of photos by the total number of users to calculate the average number of posts per user on Instagram.

Output

```
48 •    select(select count(image_url) from photos) /  
49      (select count(id) from users) as average_post;  
50  
51
```



average_post
2.5700

Average is 2.57

Findings-7

To find the bots and fake accounts :

- First, we select the **user_id** column from the **photos** table.
- Then we select the **username** column from the **users** table.
- Then, we use the **COUNT(*)** function to count the total number of likes from the **likes** table.
- Then, we **INNER JOIN** the **users** and **likes** tables on **users.id = likes.user_id** to link **users** with their respective **likes**.
- Then, by using the **GROUP BY** function, we group the output table based on **user_id**.
- We will use the **ORDER BY** function to sort the results based on total likes in descending order.

Output

```
54 •   select user_id, username, count(*) as liked_photos from likes
55     inner join users on users.id=likes.user_id
56     group by user_id
57     order by liked_photos desc;
```

The screenshot shows a MySQL query editor interface. At the top, there is a code editor with the SQL query. Below it is a toolbar with icons for back, forward, refresh, and other database operations. The main area is a "Result Grid" table with three columns: "user_id", "username", and "liked_photos". The data shows four rows of results. At the bottom of the grid, there are navigation buttons for "Result 10" and a close button.

	user_id	username	liked_photos
▶	21	Rocio33	257
	71	Nia_Haag	257
	5	Aniya_Hackett	257
	66	Mike.Auer39	257

41	Mckenna17	257
14	Jadyn81	257
57	Julien_Schmidt	257
24	Maxwell.Halvorson	257
76	Janelle.Nikolaus81	257
75	Leslie67	257
54	Duane60	257
91	Bethany20	257
36	Ollie_Ledner37	257

Analysis

Here are the key findings from the Instagram User Analytics project after performing the analysis:

- *Identified the most loyal users who have been on Instagram the longest.*

<u>id</u>	<u>username</u>	<u>created_at</u>
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

- *Out of the 100 total users, 26 users are inactive and have never posted any content, including photos, videos, or text. The Instagram marketing team should focus on re-engaging these inactive users by sending reminders, personalized notifications and content sharing on the platform.*
- *The user Kaley9 with user_id 30 is the winner of the contest, as their photo received the highest number of likes (41).*
- *The top five most commonly used hashtags are #smile (59), #beach (42), #party (39), #fun (38), and #concert (24).*
- *Most users registered on Thursday and Sunday (16 users each), making these the best days to launch ad campaigns for maximum reach and engagement.*
- *There are a total of 257 photos in the photos table and 100 users in the users table, resulting in an average of 2.57 posts per user on Instagram ($257/100 = 2.57$).*

- *Out of the total user IDs, 13 users have liked every single post on Instagram, which is not practically possible. These accounts are likely BOTS or Fake Accounts, and Instagram should consider flagging them for further review.*

5 WHY Approach for Instagram User Analytics Project

- *Why was this project conducted?*
 - *To analyze user interactions and engagement on Instagram to derive valuable insights for business growth.*
- *Why is understanding user engagement important?*
 - *It helps identify active and inactive users, popular content trends, and opportunities to improve user retention and experience.*
- *Why do we need to identify inactive users and bots?*
 - *Inactive users affect platform engagement, and bots can distort metrics, leading to inaccurate business decisions.*
- *Why should Instagram optimize marketing strategies based on user trends?*
 - *Understanding peak registration days, popular hashtags, and engagement patterns allows for better ad targeting and content recommendations.*
- *Why does this analysis benefit Instagram in the long run?*
 - *By optimizing engagement strategies, reducing bot activity, and enhancing user experience, Instagram can drive more authentic interactions, increase retention, and boost revenue.*

Conclusions

In conclusion, I would like to emphasize that not only Instagram but many other social media and commercial firms utilize such analytics to extract meaningful insights from customer data. This helps businesses identify valuable customers who contribute to growth (assets) versus inactive or fraudulent users (liabilities). Regular analysis—conducted weekly, monthly, quarterly, or yearly—enables companies to optimize engagement strategies, enhance user experience, and maximize profits while minimizing costs. Effective data-driven decision-making ensures that businesses stay competitive and improve their long-term sustainability.