Instagram User Analytics

Project Description:
This project focuses on analyzing user behavior on Instagram to provide insights for marketing and investor assessments. By using SQL fundamentals, we will answer specific questions from the marketing and management teams.
In the marketing section, we will find the oldest users, identify users who haven't posted photos, determine the winner of a contest, suggest popular hashtags, and provide insights on the best day to launch AD campaigns.
In the investor metrics section, we will analyze user engagement by calculating the average number of posts per user and assess the presence of fake accounts by identifying users who have liked every single photo.
Through this project, we aim to provide valuable data-driven recommendations to support marketing campaigns, inform decision-making, and evaluate Instagram's performance and authenticity compared to other platforms.

Approach:
As an individual working on this project, I followed a structured approach to analyze user behavior on Instagram and find meaningful insights. I began by carefully examining the provided database and familiarizing myself with its structure. Then, I utilized SQL fundamentals to retrieve the necessary information for each task, employing appropriate queries and functions. I focused on data accuracy and quality throughout the project, ensuring reliable results. By leveraging my SQL skills and maintaining a systematic workflow, I successfully executed the project and created a comprehensive report that fulfilled the objectives of providing marketing insights and investor metrics.

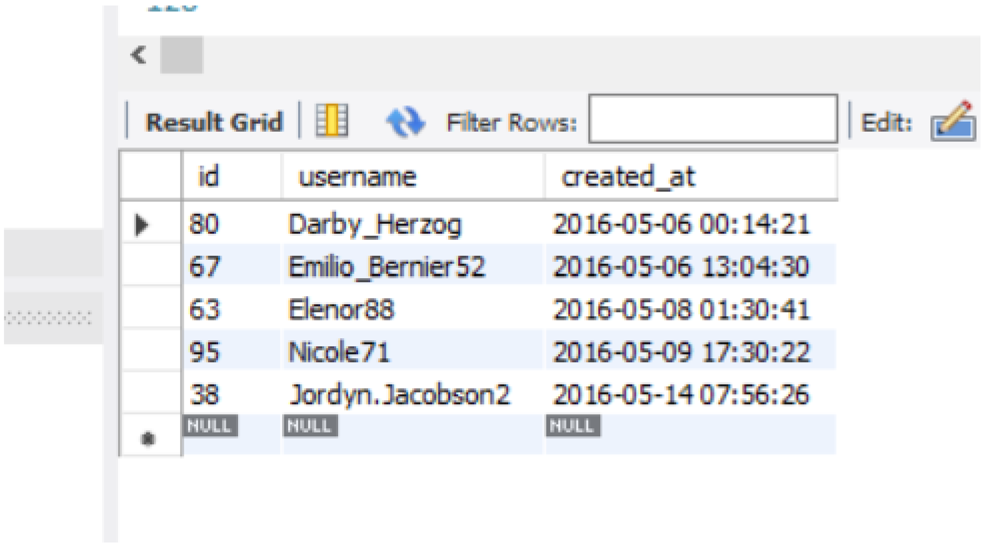
Tech-Stack Used:
For this project, I utilized MySQL Workbench 8.0 as the primary software tool. MySQL Workbench is an integrated development environment (IDE) for MySQL databases, providing a graphical interface for designing, querying, and managing databases.

Insights:

1. Rewarding Most Loyal Users: Identifying the oldest users of Instagram helps recognize and reward long-term user loyalty.
For finding most loyal users following query was used:

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

Here we get the information of the oldest users from the "users" table, including their ID, username, and the date they joined, sorted in order of their joining date, we limit it to 5, to get top 5 users.
Following is the result of query:



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

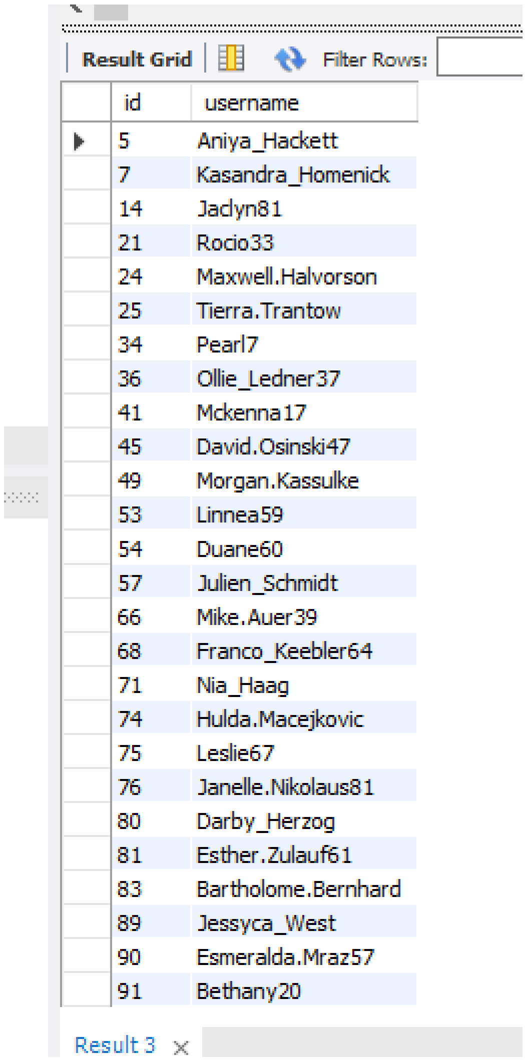
2. Remind Inactive Users to Start Posting: Users who have never posted a single photo on Instagram represent an opportunity for re-engagement through targeted promotional emails.

For finding inactive users following query was used:

```
SELECT users.id, users.username
FROM users
LEFT JOIN photos ON users.id = photos.user_id
WHERE photos.id IS NULL;
```

Here we select the ID and username from the "users" table, where we find users who have not posted any photos based on the left join with the "photos" table, matching the user IDs. We filter out users who have a null value for the photo ID, indicating that they haven't posted any photos.

Following is the result of query:

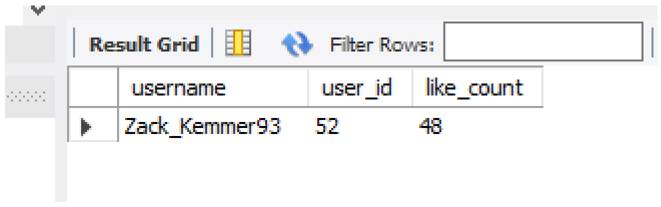


id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

3. Declaring Contest Winner: The winner of a contest can be determined by the user with the most likes on a single photo, ensuring a fair and accurate declaration.
For finding user with most likes on a photo following query was used:

```
SELECT u.username, p.user_id, l.like_count
FROM (
    SELECT photo_id, COUNT(*) AS like_count
    FROM likes
    GROUP BY photo_id
    ORDER BY like_count DESC
    LIMIT 1
) l
JOIN photos p ON l.photo_id = p.id
JOIN users u ON p.user_id = u.id;
```

Here we are finding the username, user ID, and number of likes for the user with the most popular photo. We count the likes for each photo, find the photo with the highest count, and then match it with the user who owns the photo. By joining the photos, users and likes table, we retrieve the username and user ID for the user with the most likes on their photo.
Following is the result of query:

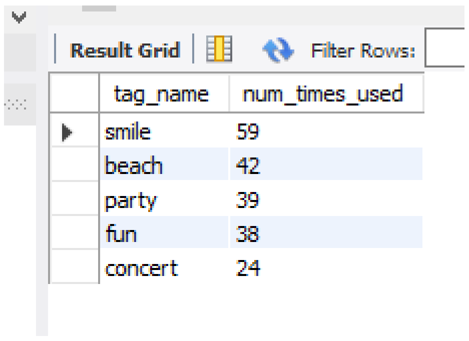


username	user_id	like_count
Zack_Kemmer93	52	48

4. Hashtag Researching: Identifying the top five most commonly used hashtags on Instagram allows for effective hashtag selection to reach a broader audience.
For finding top 5 hashtags used most commonly, following query was used:

```
SELECT t.tag_name, COUNT(*) AS num_times_used
FROM tags t
JOIN photo_tags pt ON t.id = pt.tag_id
GROUP BY t.tag_name
ORDER BY num_times_used DESC
LIMIT 5;
```

To find the top 5 most commonly used hashtags, we join the "tags" table with the "photo_tags" table using their respective IDs. By grouping the records based on the tag name, we count the number of times each tag has been used. We then sort the results in descending order based on the count of usage, selecting only the top 5.
Following is the result of query:

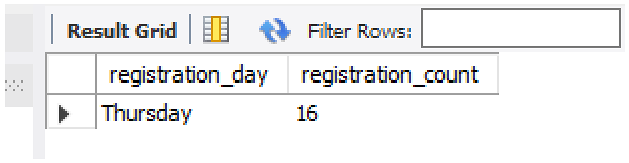


tag_name	num_times_used
smile	59
beach	42
party	39
fun	38
concert	24

5. Launch AD Campaign: Analyzing user registration patterns reveals the best day to launch advertisements, maximizing their potential impact and reach.
For finding day on which most users register, following query was used:

```
SELECT DAYNAME(created_at) AS registration_day, COUNT(*) AS registration_count
FROM users
GROUP BY registration_day
ORDER BY registration_count DESC
LIMIT 1;
```

By selecting the day name from the "created_at" column of the "users" table, we group the registrations by day of the week. We count the number of registrations for each day and sort the results in descending order based on the registration count. We select only the top result, which represents the day of the week with the highest number of user registrations on Instagram.
Following is the result of query:

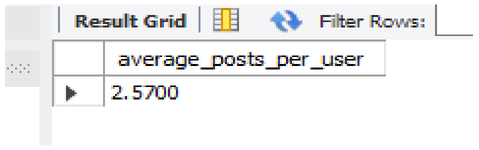


registration_day	registration_count
Thursday	16

6. User Engagement: Calculating the average number of posts per user provides insights into user activity levels and potential trends in engagement.
For finding average number of posts per user, following query was used:

```
SELECT COUNT(photos.id) / COUNT(DISTINCT users.id) AS average_posts_per_user
FROM users
LEFT JOIN photos ON users.id = photos.user_id;
```

By joining the "users" table with the "photos" table using their respective IDs, we count the total number of photos. We also count the distinct number of users. Then, we divide the total number of photos by the distinct number of users to calculate the average posts per user. This gives an average number of posts per user.
Following is the result of query:



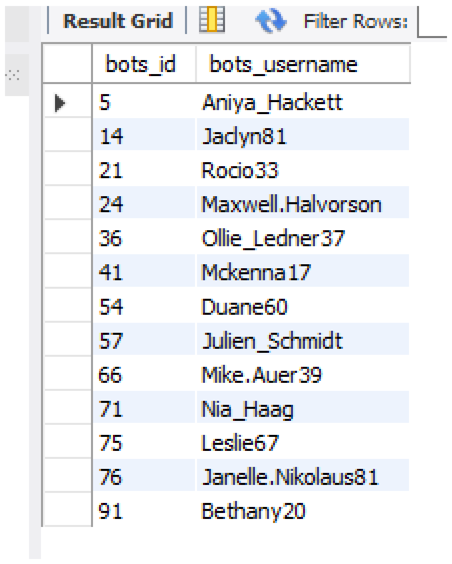
average_posts_per_user
2.5700

7. Bots & Fake Accounts: Identifying users who have liked every single photo helps detect the presence of bots and fake accounts, ensuring a more authentic user community.

```
SELECT u.id as bots_id, u.username AS bots_username
FROM (
   SELECT l.user_id
   FROM (
      SELECT p.id
      FROM photos p
   ) AS all_photos
   LEFT JOIN likes l ON all_photos.id = l.photo_id
   GROUP BY l.user_id
   HAVING COUNT(DISTINCT l.photo_id) = (SELECT COUNT(*) FROM photos)
) AS bots
JOIN users u ON bots.user_id = u.id;
```

To find bots, we find the users who have liked every single photo on Instagram. First, we select all the photo IDs from the "photos" table. Next, we join this list of photo IDs (stored as "all_photos") with the "likes" table on the photo ID to find the corresponding user IDs who have liked those photos. We group the results by the user ID and filter them using the "HAVING" clause, where we count the distinct photo IDs liked by each user and compare it to the total count of photos in the "photos" table. Then, we join this result (stored as "bots") with the "users" table using the matching user IDs.

Following is the result of query:



bots_id	bots_username
5	Aniya_Hackett
14	Jaclyn81
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

Results:

While working on this project, I have gained a better understanding of user analytics and SQL fundamentals. By analyzing user data on Instagram, I was able to provide insights on various aspects such as rewarding loyal users, identifying inactive users, declaring contest winners, researching popular hashtags, determining the best day to launch ad campaigns, assessing user engagement, and detecting fake accounts.

This project has helped me enhance my SQL skills, particularly in querying and manipulating data to derive meaningful insights. It has also improved my ability to interpret data and provide actionable recommendations based on the analysis. Overall, this project has deepened my understanding of user behavior analysis and its application in making informed decisions for product development and marketing strategies.