Submitted by: Asmita Singh

Instagram User Analysis

Project Description:

This project focuses on analysing Instagram user data to uncover valuable insights that can help the platform grow. By studying user engagement and behaviour, we aim to understand how users interact with Instagram. These insights will be beneficial for multiple teams:

- The marketing team can use the data to launch effective Instagram campaigns.
- The product team can prioritize the development of new Instagram features.
- The development team can enhance the overall Instagram user experience.

Using SQL and MySQL Workbench, we will analyse Instagram data and address questions from the management team. The insights gained will support the product manager and other teams in making informed decisions about the future of Instagram. This project emphasizes the importance of leveraging data to drive Instagram's growth and success.

SQL Task:

A) Marketing Analysis

- Loyal User Reward: The marketing team plans to recognize and reward Instagram's most loyal users, specifically those who have been on the platform the longest.
  - Identify the top five oldest users on Instagram from the database.
- Inactive User Engagement: To re-engage users who are not active, the team wants to send promotional emails encouraging them to start posting.
  - Identify users who have never posted a single photo on Instagram.
- Contest Winner Declaration: A contest was organized, and the winner is the user whose single photo has received the most likes.
  - Find the contest winner and share their details with the team.
- Hashtag Research**:** A partner brand wants to optimize their posts by using the most effective hashtags to maximize reach.
  - Identify and recommend the top five most popular hashtags on Instagram.
- Ad Campaign Launch: The team needs to know the ideal day to launch ad campaigns.
  - Analyze the data to determine the day of the week with the highest user registrations on Instagram and provide insights for scheduling ad campaigns.

B) Investor Metrics

- User Engagement: Investors are interested in understanding whether users are actively posting on Instagram or if there has been a decline in posting activity.
  - Calculate the average number of posts per user on Instagram. Additionally, compute the total number of photos on Instagram divided by the total number of users.
- Bots & Fake Accounts: Investors are concerned about the presence of fake or dummy accounts on the platform.
  - *Identify potential bot accounts by finding users who have liked every single photo on Instagram, as this behaviour is not typical for a genuine user*.

Approach:

Setup and Preparation

- I will load database creation script into MySQL Workbench.
- Execute the script to create necessary tables and populate them with data.
- Verify the database structure and data by exploring the tables and columns.

Understand the Requirements

- I will identify the tables and columns required to address the tasks.
- Then will document the key queries to be written for analysis, ensuring a clear goal for each.

Perform Data Analysis

- Task 1: Loyal User Reward

  - ➢ Will write an SQL query to sort users by account creation date in ascending order.
  - ➢ Select the top five users based on the oldest creation dates.

- Task 2: Inactive User Engagement

  - ➢ Will write a query to identify users who have never posted a photo.
  - ➢ Filter records where the post count is zero.

- Task 3: Contest Winner Declaration

  - ➢ Will write a query to find the photo with the highest number of likes.
  - ➢ Retrieve the associated user details for the photo.

- Task 4: Hashtag Research
  - ➢ Will query the database to count occurrences of each hashtag.
  - ➢ Identify the top five most frequently used hashtags.

- Task 5: Ad Campaign Launch
  - ➢ Will analyze user registration dates to determine the day of the week with the most registrations.
  - ➢ Group data by day and order by registration counts.

- Task 6: User Engagement
  - ➢ Will calculate the average number of posts per user by dividing the total number of posts by the total number of users.
  - ➢ Compute the ratio of total photos to total users.

- Task 7: Bots & Fake Accounts
  - ➢ Will identify users who have liked every single photo on Instagram by comparing the number of likes per user with the total number of photos.

Document the Findings

- After all this, I will organize the results of my analysis into a clear, structured format.

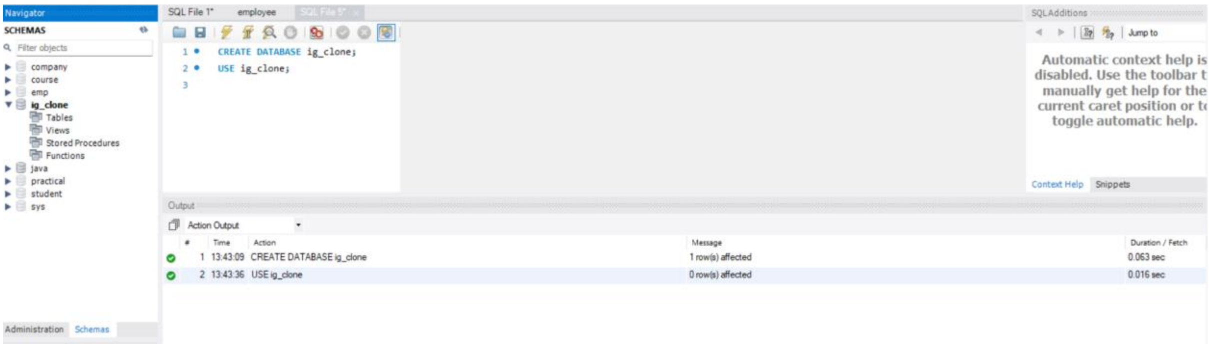
Tech Stack used:

- MySQL Workbench
- Database: https://docs.google.com/document/d/1-WhNRX1iYJIz7e5l28DMPWgsPklpE_w6/edit

Submitted by: Asmita Singh

Working:

Creating and using Data base

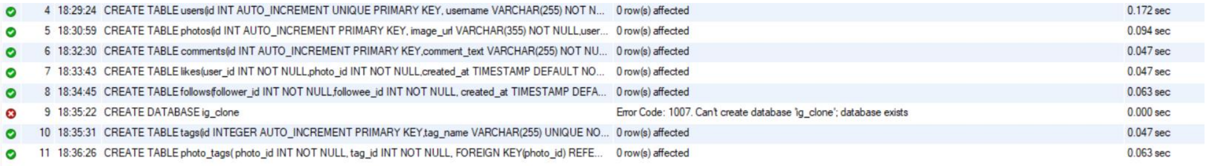


```
CREATE DATABASE ig_clone;
USE ig_clone;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

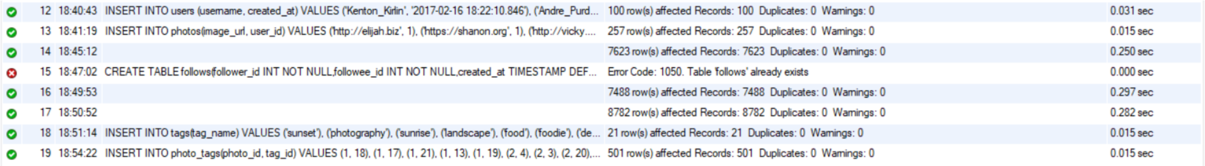
#	Time	Action	Message	Duration / Fetch
1	13:43:09	CREATE DATABASE ig_clone	1 row(s) affected	0.063 sec
2	13:43:36	USE ig_clone	0 row(s) affected	0.016 sec

Created tables users, photos, comments, likes, follows, tags, and photo_tags



4	18:29:24	CREATE TABLE users(id INT AUTO_INCREMENT UNIQUE PRIMARY KEY, username VARCHAR(255) NOT N...	0 row(s) affected	0.172 sec
5	18:30:59	CREATE TABLE photos(id INT AUTO_INCREMENT PRIMARY KEY, image_url VARCHAR(355) NOT NULL,user...	0 row(s) affected	0.094 sec
6	18:32:30	CREATE TABLE comments(id INT AUTO_INCREMENT PRIMARY KEY,comment_text VARCHAR(255) NOT NU...	0 row(s) affected	0.047 sec
7	18:33:43	CREATE TABLE likes(user_id INT NOT NULL,photo_id INT NOT NULL,created_at TIMESTAMP DEFAULT NO...	0 row(s) affected	0.047 sec
8	18:34:45	CREATE TABLE follows(follower_id INT NOT NULL,followee_id INT NOT NULL, created_at TIMESTAMP DEFA...	0 row(s) affected	0.063 sec
9	18:35:22	CREATE DATABASE ig_clone	Error Code: 1007. Can't create database 'ig_clone'; database exists	0.000 sec
10	18:35:31	CREATE TABLE tags(id INTEGER AUTO_INCREMENT PRIMARY KEY,tag_name VARCHAR(255) UNIQUE NO...	0 row(s) affected	0.047 sec
11	18:36:26	CREATE TABLE photo_tags(photo_id INT NOT NULL, tag_id INT NOT NULL, FOREIGN KEY(photo_id) REFE...	0 row(s) affected	0.063 sec

Inserted values in the tables:

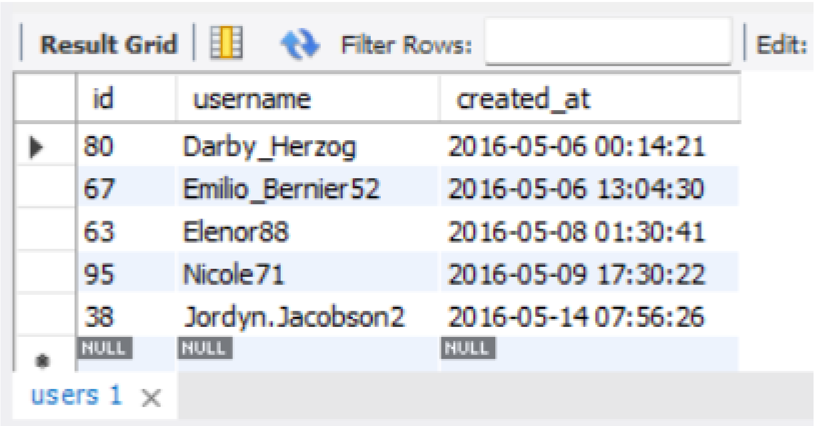


12	18:40:43	INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purd...	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.031 sec
13	18:41:19	INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky....	257 row(s) affected Records: 257 Duplicates: 0 Warnings: 0	0.015 sec
14	18:45:12		7623 row(s) affected Records: 7623 Duplicates: 0 Warnings: 0	0.250 sec
15	18:47:02	CREATE TABLE follows(follower_id INT NOT NULL,followee_id INT NOT NULL,created_at TIMESTAMP DEF...	Error Code: 1050. Table 'follows' already exists	0.000 sec
16	18:49:53		7488 row(s) affected Records: 7488 Duplicates: 0 Warnings: 0	0.297 sec
17	18:50:52		8782 row(s) affected Records: 8782 Duplicates: 0 Warnings: 0	0.282 sec
18	18:51:14	INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('de...	21 row(s) affected Records: 21 Duplicates: 0 Warnings: 0	0.015 sec
19	18:54:22	INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20),...	501 row(s) affected Records: 501 Duplicates: 0 Warnings: 0	0.015 sec

Results:

- <u>Task 1: Identify the top five oldest users on Instagram from the database.</u>

  SQL- SELECT id, username, created_at FROM users ORDER BY created_at ASC LIMIT 5;

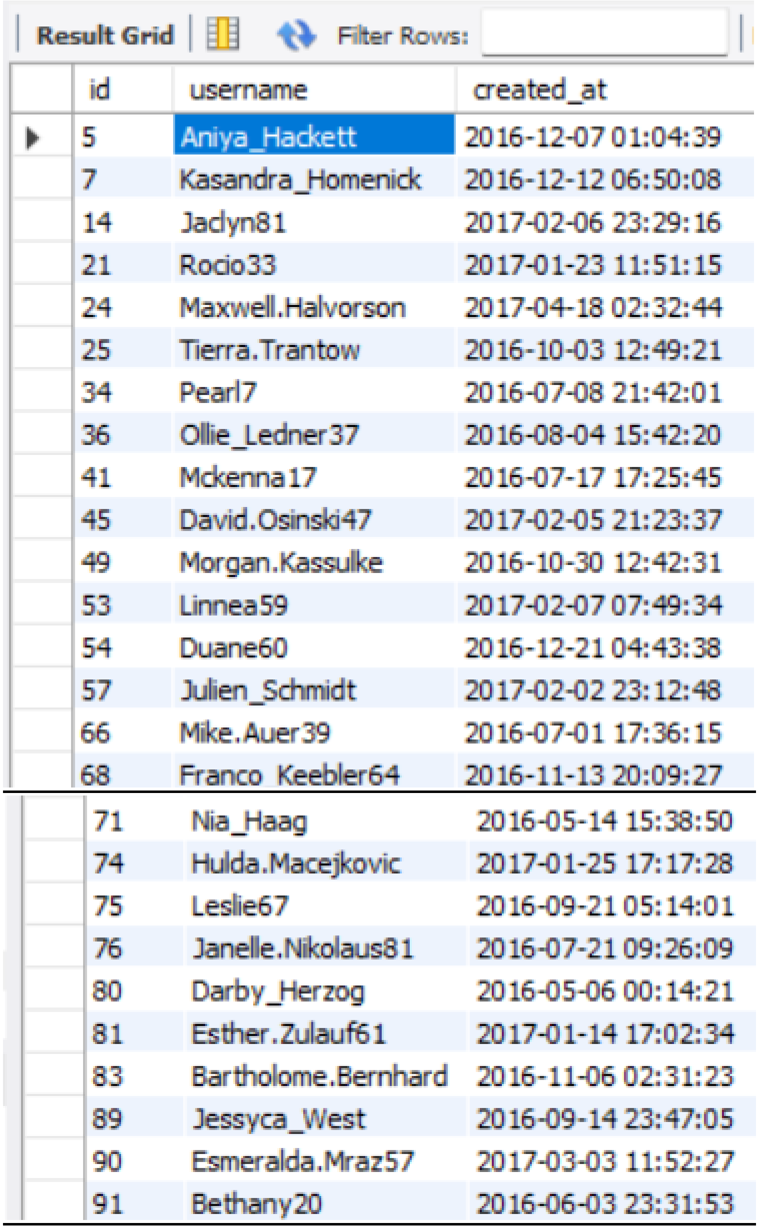


id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

Submitted by: Asmita Singh

- Task 2: Identify users who have never posted a single photo on Instagram.

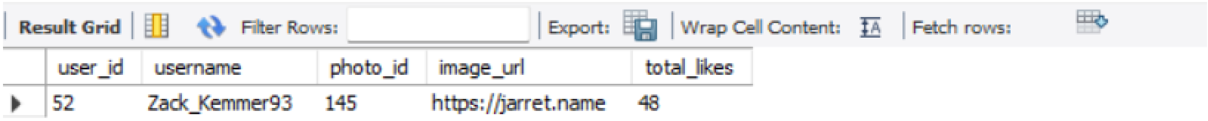
SQL- SELECT u.id, u.username, u.created_at FROM users u LEFT JOIN photos p ON u.id = p.user_id WHERE p.id IS NULL;



id	username	created_at
5	Aniya_Hackett	2016-12-07 01:04:39
7	Kasandra_Homenick	2016-12-12 06:50:08
14	Jaclyn81	2017-02-06 23:29:16
21	Rocio33	2017-01-23 11:51:15
24	Maxwell.Halvorson	2017-04-18 02:32:44
25	Tierra.Trantow	2016-10-03 12:49:21
34	Pearl7	2016-07-08 21:42:01
36	Ollie_Ledner37	2016-08-04 15:42:20
41	Mckenna17	2016-07-17 17:25:45
45	David.Osinski47	2017-02-05 21:23:37
49	Morgan.Kassulke	2016-10-30 12:42:31
53	Linnea59	2017-02-07 07:49:34
54	Duane60	2016-12-21 04:43:38
57	Julien_Schmidt	2017-02-02 23:12:48
66	Mike.Auer39	2016-07-01 17:36:15
68	Franco_Keebler64	2016-11-13 20:09:27
71	Nia_Haag	2016-05-14 15:38:50
74	Hulda.Macejkovic	2017-01-25 17:17:28
75	Leslie67	2016-09-21 05:14:01
76	Janelle.Nikolaus81	2016-07-21 09:26:09
80	Darby_Herzog	2016-05-06 00:14:21
81	Esther.Zulauf61	2017-01-14 17:02:34
83	Bartholome.Bernhard	2016-11-06 02:31:23
89	Jessyca_West	2016-09-14 23:47:05
90	Esmeralda.Mraz57	2017-03-03 11:52:27
91	Bethany20	2016-06-03 23:31:53

- Task 3: Find the contest winner and share their details with the team.

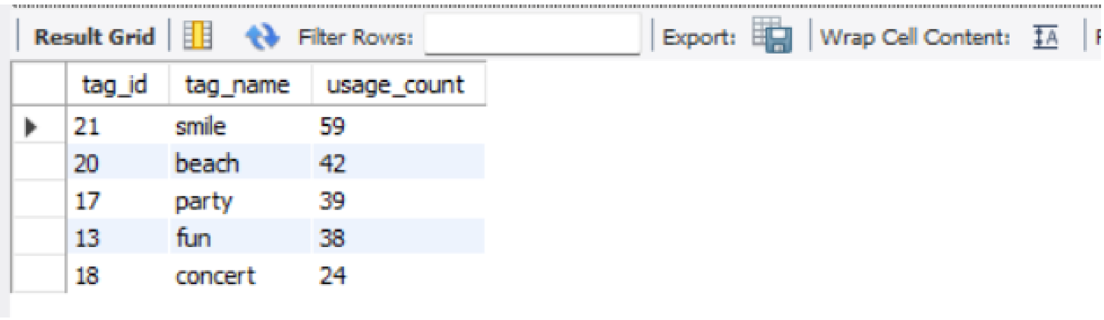
SQL- SELECT u.id AS user_id, u.username, p.id AS photo_id, p.image_url, COUNT(l.user_id) AS total_likes FROM photos p JOIN likes l ON p.id = l.photo_id JOIN users u ON p.user_id = u.id GROUP BY p.id ORDER BY total_likes DESC LIMIT 1;



user_id	username	photo_id	image_url	total_likes
52	Zack_Kemmer93	145	https://jarret.name	48

- Task 4: Identify and recommend the top five most popular hashtags on Instagram.

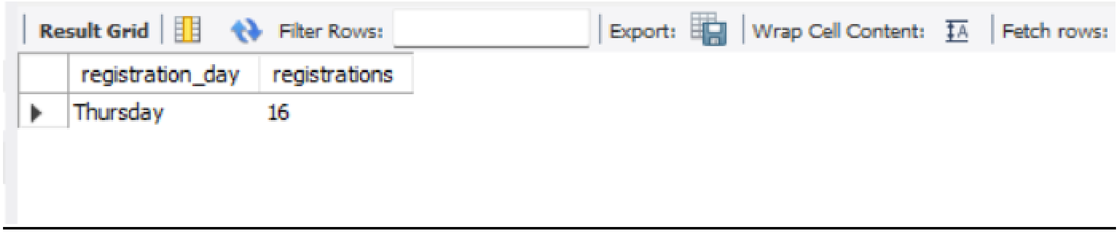
  SQL- SELECT t.id AS tag_id, t.tag_name, COUNT (pt.photo_id) AS usage_count FROM tags t JOIN photo_tags pt ON t.id = pt.tag_id GROUP BY t.id, t.tag_name ORDER BY usage_count DESC LIMIT 5;



tag_id	tag_name	usage_count
21	smile	59
20	beach	42
17	party	39
13	fun	38
18	concert	24

- Task 5: Analyze the data to determine the day of the week with the highest user registrations on Instagram and provide insights for scheduling ad campaigns.

  SQL- SELECT DAYNAME(created_at) AS registration_day, COUNT(id) AS registrations FROM users GROUP BY registration_day ORDER BY registrations DESC LIMIT 1;

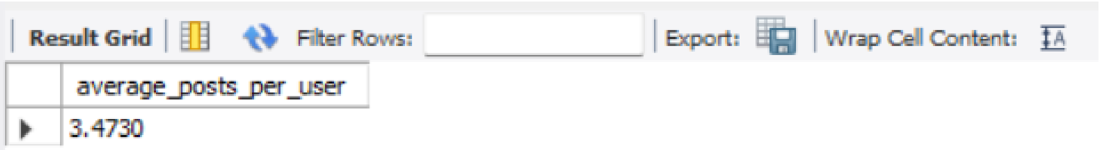


registration_day	registrations
Thursday	16

- Task 6: Calculate the average number of posts per user on Instagram. (Additionally, compute the total number of photos on Instagram divided by the total number of users.)

  1. Average Number of Posts per User:

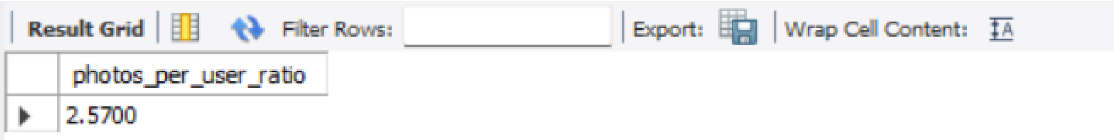
  SQL- SELECT AVG(post_count) AS average_posts_per_user FROM (SELECT user_id, COUNT(id) AS post_count FROM photos GROUP BY user_id) AS user_posts;



average_posts_per_user
3.4730

  2. Total Number of Photos Divided by the Total Number of Users:

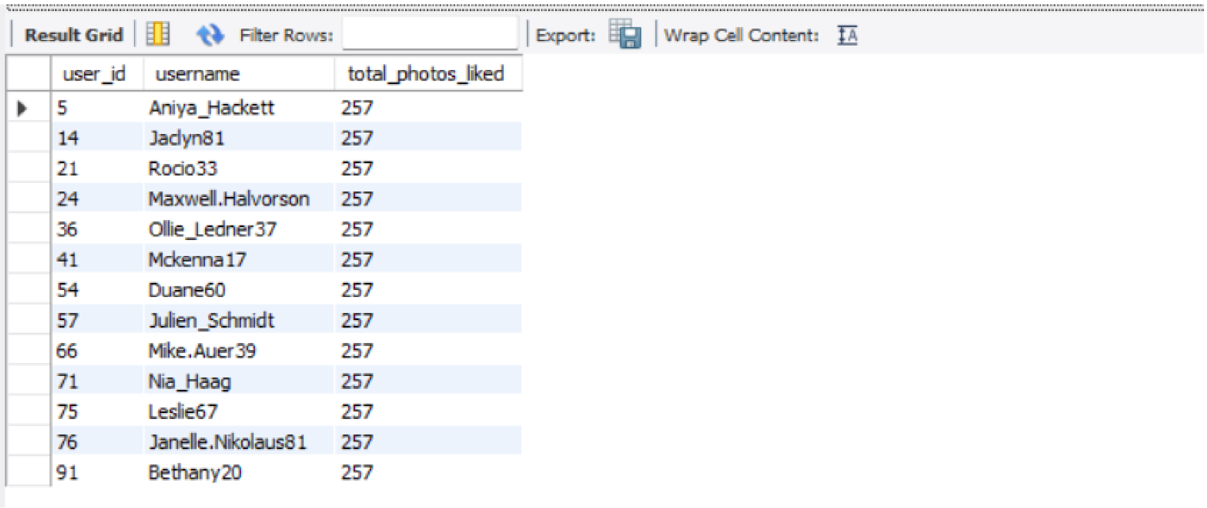
  SQL- SELECT (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS photos_per_user_ratio;



photos_per_user_ratio
2.5700

- Task 7: Identify potential bot accounts by finding users who have liked every single photo on Instagram, as this behaviour is not typical for a genuine user.

  SQL- SELECT u.id AS user_id, u.username, COUNT(p.id) AS total_photos_liked FROM users u JOIN likes l ON u.id = l.user_id JOIN photos p ON l.photo_id = p.id GROUP BY u.id HAVING total_photos_liked = (SELECT COUNT(*) FROM photos);



user_id	username	total_photos_liked
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

Insights:

From Task 1, I learned how relational databases work, specifically how tables are linked via foreign keys and how data can be joined across different tables using JOIN operations. This taught me how to manage and query data spread across multiple tables while ensuring data integrity and efficient querying. In Task 2, I learned how to retrieve specific subsets of data using SELECT queries, filter them with WHERE conditions, and apply GROUP BY and HAVING clauses. This helped me understand how to extract relevant data and summarize it effectively, especially in the context of real-world business cases like analyzing user activity. Task 3 taught me how to calculate the average, COUNT, and use other aggregation functions to summarize data. This was helpful in tasks such as determining the average number of posts per user and identifying the most popular hashtags. I became comfortable using functions like COUNT(), AVG(), and SUM(), which are essential for deriving insights from large datasets. Through Task 4, I learned how to handle timestamps and extract meaningful information based on date and time, like identifying the day with the highest user registrations. I became familiar with using date functions such as DAYNAME(), which is crucial when working with time-based data, like tracking user behaviour over specific periods. In Task 5, I learned to use the HAVING clause, which allowed me to filter groups based on aggregate values, such as detecting bot accounts. This taught me how to go beyond simple row-level filtering with WHERE and apply more advanced filtering in aggregated data. In Task 6, I gained experience in identifying abnormal patterns in the data, such as inactive users and potential bot accounts. I learned to apply business logic to spot outliers and anomalies, which is crucial for data cleaning and ensuring the quality of data used for analysis. Through all the tasks, I learned how to apply my SQL skills to real business scenarios like user engagement, contest winners, and hashtag research. This helped me understand how to convert business requirements into SQL queries, improving my problem-solving abilities and bridging the gap between theoretical knowledge and practical application.

Overall, as I worked with larger datasets in some of the tasks, I also learned the importance of optimizing queries for performance. This involved writing efficient SQL queries that scale well with data growth, an essential skill for working with large databases like using aggregate functions, using the keyword LIMIT, using joints and filtering data using GROUP BY and HAVING clause.

Conclusion

During this project, I learned how to use SQL to solve real business problems. I understood how relational databases work, how data is organized in different tables, and how to write SQL queries to get the data I need. I also learned how to use functions to summarize data, filter it, and find patterns. I got hands-on experience with tasks like identifying bot accounts, finding inactive users, and analyzing user behaviour over time. This helped me improve my problem-solving skills and apply SQL to real situations. Additionally, I learned the importance of making queries run efficiently, especially when working with large amounts of data. Overall, this project improved my SQL skills and gave me a better understanding of how to turn data into useful insights, which will help me with future data analysis tasks.