TRAINITY PROJECT 2 - Instagram User Analytics

-SQL FUNDAMENTALS

Project Description: The Instagram User Analytics project involves using SQL and MySQL Workbench in finding the insights from the collected user engagement with Instagram details tables of ig_clone datatbase. By understanding how the users interact with Instagram, various and prominent data-driven decisions can be made such that user experience will be enhanced, launch effective marketing campaigns and new features.

Approach:

Step 1 -Understand the Requirement : Clearly need to understand what data need to be extracted from the database.

Step 2 - Choose Tables needed from the database: Find the tables in database which contain the required data to be extracted.

Step 3 – Extract the Required data: The SQL statements used for this project as per the use case:

i. Select: Using 'Select' statement to extract the required data.
ii. Where: Using 'Where' clause to filter the data based on condition.
iii. Aggregated Functions: COUNT(), SUM(), AVG(), MIN(),MAX() are used to simply aggregate the data.
iv. GROUP BY & ORDER BY: Using GROUP BY to categorize the data and ORDER BY to sort the data.
v. JOINS: Using Joins to join the relevant tables when needed.
vi. Window Functions: Window functions return value for each and every row while still considering the group as a whole. For example Rank() assigns rank to each row within a partition, with ties, leaving a gap in ranking.

Step 4 – Check query for accuracy and efficiency : Ensure the query extracts the correct(required) data and runs efficiently without unnecessary computations.

Tech-Stack Used:

1. **MySQL Workbench 8.0 CE:**

MySQL Workbench provides a user-friendly graphical interface that simplifies database management tasks, including data modeling and query execution. This helps in quickly visualizing and understanding the structure of the Instagram user data. It supports window functions, which play important role in deriving in-depth insights into user behaviour and engagement with Instagram.

2. **MySQL Server 8.0:**
MySQL Server 8.0 offers robust performance, scalability, and reliability, which are crucial for handling extensive user data and performing complex queries without reducing speed or accuracy.

3. **SQL(Structured Query Language):** It is an easy language used to perform various operations with relational databases such as creating tables, modifying tables, extracting essential information etc., in a easy manner.

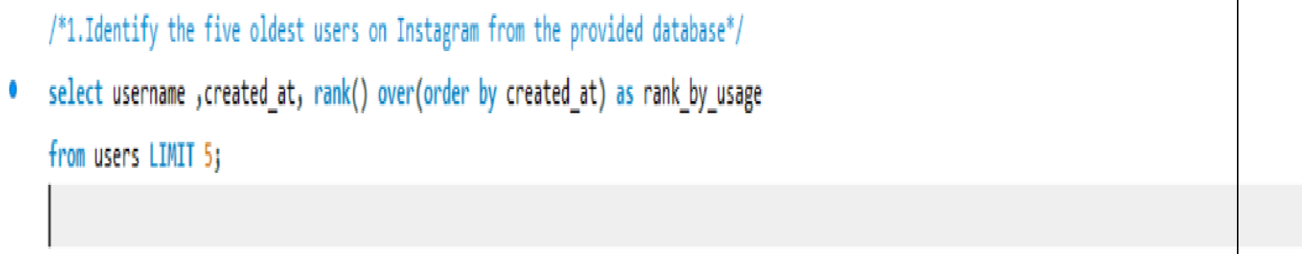
SQL Tasks :

A) Marketing Analysis:

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

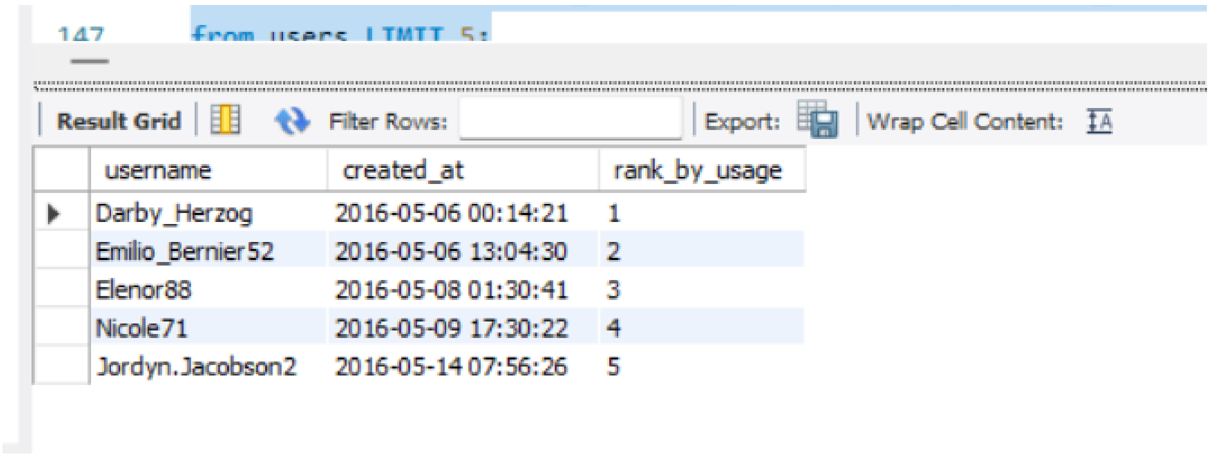
- ➔ To find five oldest users, users table is required.
- ➔ From created_at column, we can extract the account created information.
- ➔ Using window function rank() over(order by created_at), created_at column values are sorted in ascending order i.,e from the oldest years(small values) to recent years and given ranks to the rows based on that.
- ➔ LIMIT function(LIMIT 5)for the five oldest users data retrieval.

Query:



```
/*1.Identify the five oldest users on Instagram from the provided database*/
select username ,created_at, rank() over(order by created_at) as rank_by_usage
from users LIMIT 5;
```

Output:



```
from users LIMIT 5;
```

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

username	created_at	rank_by_usage
Darby_Herzog	2016-05-06 00:14:21	1
Emilio_Bernier52	2016-05-06 13:04:30	2
Elenor88	2016-05-08 01:30:41	3
Nicole71	2016-05-09 17:30:22	4
Jordyn.Jacobson2	2016-05-14 07:56:26	5

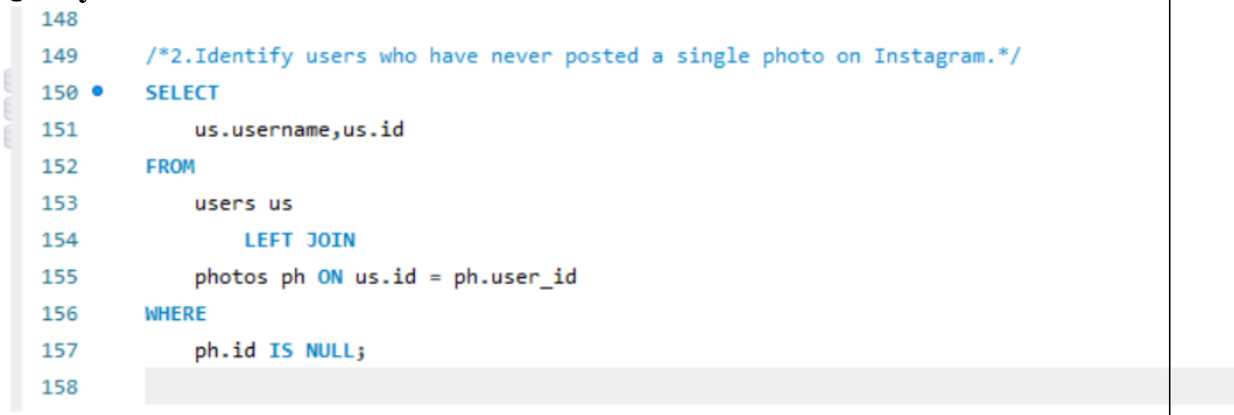
➔ From the output, we got the five oldest users of Instagram with the data of username, created_at and the rank based upon creation of account(rank_by_usage)

<u>2.Inactive User Engagement:</u> The team wants to encourage inactive users to start posting by sending them promotional emails.

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

➔ To do the task, users and photos tables are required.
➔ To combine the relevant tables, we use JOINS(Here used LEFT JOIN for getting the required data from users table(left table)) based on the common column present in both tables i.e., id from users and user_id from photos tables.
➔ The condition to be satisfied is no photo posted so the condition we provided in where clause is ph.id IS NULL.

Query:

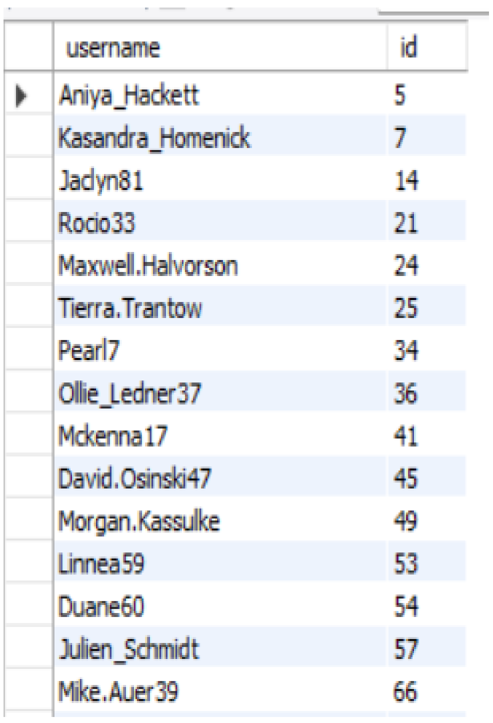
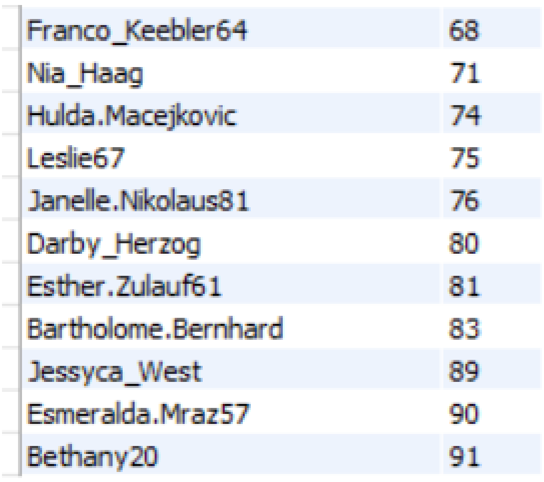


```

/*2.Identify users who have never posted a single photo on Instagram.*/
SELECT
    us.username,us.id
FROM
    users us
        LEFT JOIN
    photos ph ON us.id = ph.user_id
WHERE
    ph.id IS NULL;

```

Output:

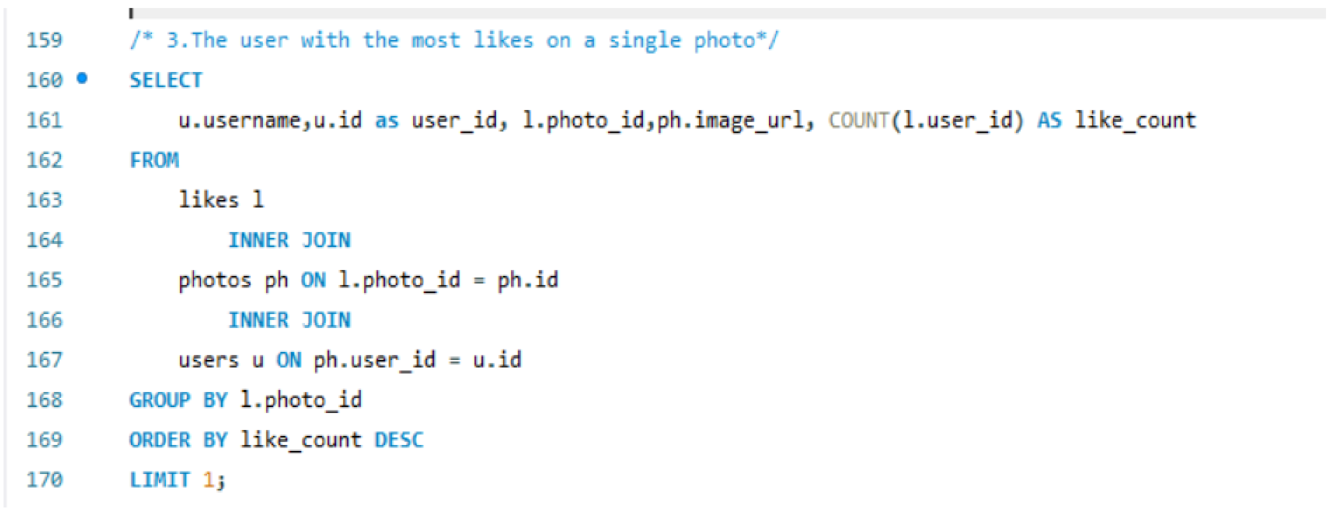
username	id
Aniya_Hackett	5
Kasandra_Homenick	7
Jaclyn81	14
Rocio33	21
Maxwell.Halvorson	24
Tierra.Trantow	25
Pearl7	34
Ollie_Ledner37	36
Mckenna17	41
David.Osinski47	45
Morgan.Kassulke	49
Linnea59	53
Duane60	54
Julien_Schmidt	57
Mike.Auer39	66
Franco_Keebler64	68
Nia_Haag	71
Hulda.Macejkovic	74
Leslie67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

- ➔ From the output, we got all the users details (username,id from users) who never posted a single photo on Instagram. There are totally **26 users** out of 100 users who have never posted a single photo on Instagram.

<u>3.Contest Winner Declaration:</u> The team has organized a contest where the user with the most likes on a single photo wins. **Task:** Determine the winner of the contest and provide their details to the team.

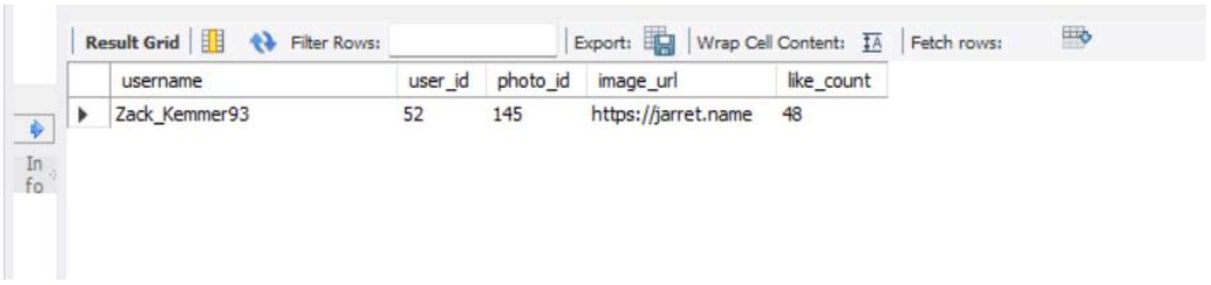
- ➔ To perform the task, we need three tables, they are : likes, photos and users
- ➔ Select the required data to be extracted from the tables and alias names are also given.
- ➔ To combine the relevant tables we perform Inner Join, which combines the equal values of common columns of two tables. Here we perform the Inner Join first on likes and photos ; then on photos and users.
- ➔ Using group by function, we will group the output based on l.photo_id and using order by function, we will sort the output based on like_count(count(l.user_id)) in descending order.
- ➔ LIMIT function(LIMIT 1) to find out the user with the most likes on single photo.

Query:



```
/* 3.The user with the most likes on a single photo*/
SELECT
    u.username,u.id as user_id, l.photo_id,ph.image_url, COUNT(l.user_id) AS like_count
FROM
    likes l
        INNER JOIN
    photos ph ON l.photo_id = ph.id
        INNER JOIN
    users u ON ph.user_id = u.id
GROUP BY l.photo_id
ORDER BY like_count DESC
LIMIT 1;
```

Output:



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

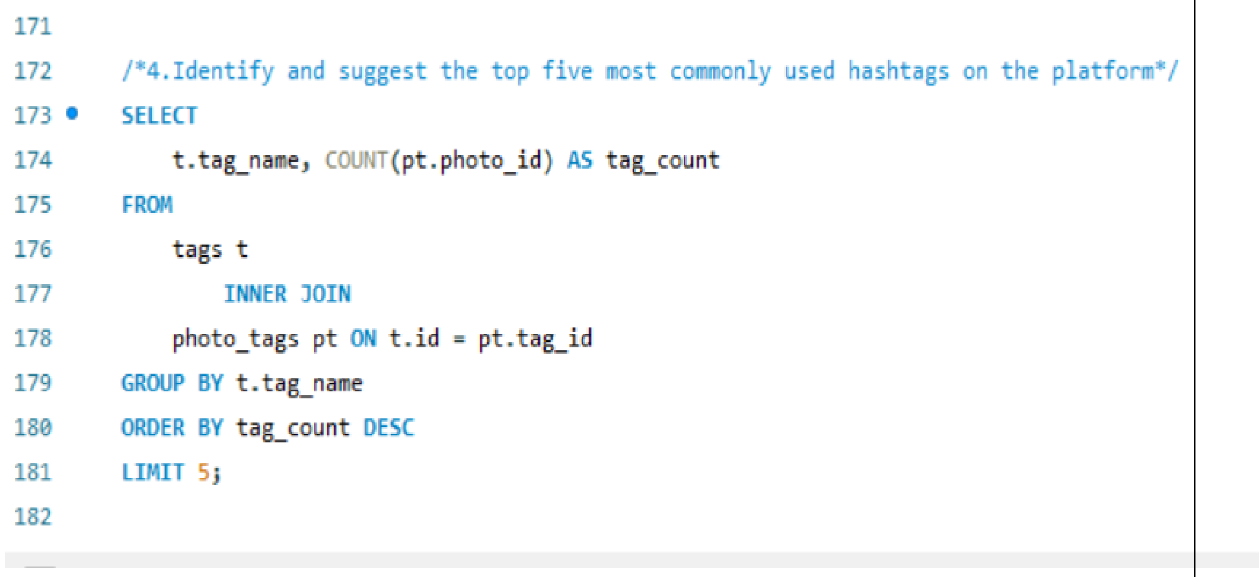
- ➔ From the output, the winner's details are found out! The winner with the most likes on single photo is **Zack_Kemmer93**(username) with user_id :52, photo_id :145, image_url given with like_count: **48**.

  <u>4.Hashtag Research</u>: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people. **Task:** Identify and suggest the top five most commonly used hashtags on the platform.

- ➔ To perform the task, we need tags and photo_tags tables.
- ➔ Select the required data to be extracted using select statement and aliases can also be given to the column names and tables.
- ➔ Combine the relevant tables(tags and photo_tags) using INNER JOIN based on common column i.e., t.id and pt.tag_id
- ➔ Using group by function, group the output based on t.tag_name and using order by function, we will sort the output based on tag_count in descending order.

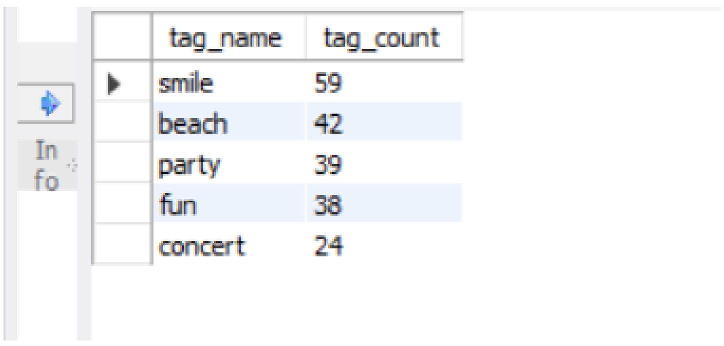
➔ Using LIMIT function(LIMIT 5), extract the details of top five most commonly used hashtags on Instagram.

Query:



```
/*4.Identify and suggest the top five most commonly used hashtags on the platform*/
SELECT
    t.tag_name, COUNT(pt.photo_id) AS tag_count
FROM
    tags t
        INNER JOIN
    photo_tags pt ON t.id = pt.tag_id
GROUP BY t.tag_name
ORDER BY tag_count DESC
LIMIT 5;
```

Output:



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

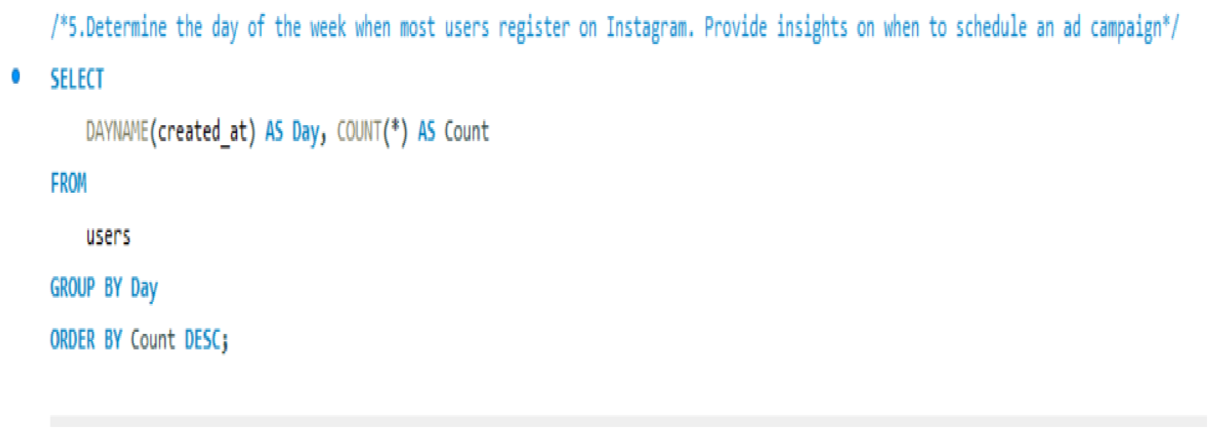
➔ From the output, tag_name and tag_count of the top five most commonly used hashtags on Instagram are extracted**(smile,beach,party,fun,concert)**

<u>5.Ad Campaign Launch:</u> The team wants to know the best day of the week to launch ads.
Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

➔ To perform the task, we define the required columns of the output and create the aliases using select DAYNAME(created_at) AS Fay, COUNT(*) AS Count FROM users. The only table required here is users.

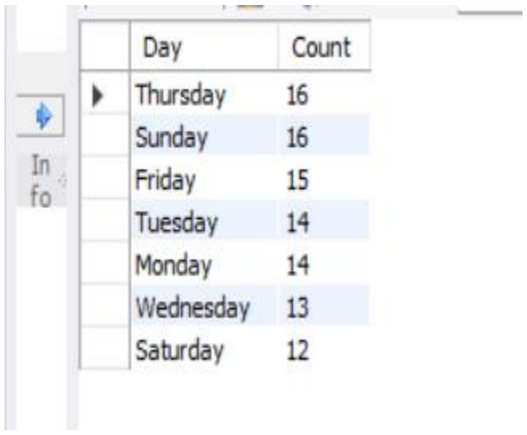
➔ Using group by function, group the output based on Day and using order by function, we will sort the output based on Count in descending order.

Query:



```
/*5.Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign*/
SELECT
    DAYNAME(created_at) AS Day, COUNT(*) AS Count
FROM
    users
GROUP BY Day
ORDER BY Count DESC;
```

Output:



Day	Count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

➔ From the output we can say that, On both **Thursday and Sunday**, most users are registered on Instagram with count 16 each.

B) Investor Metrics:

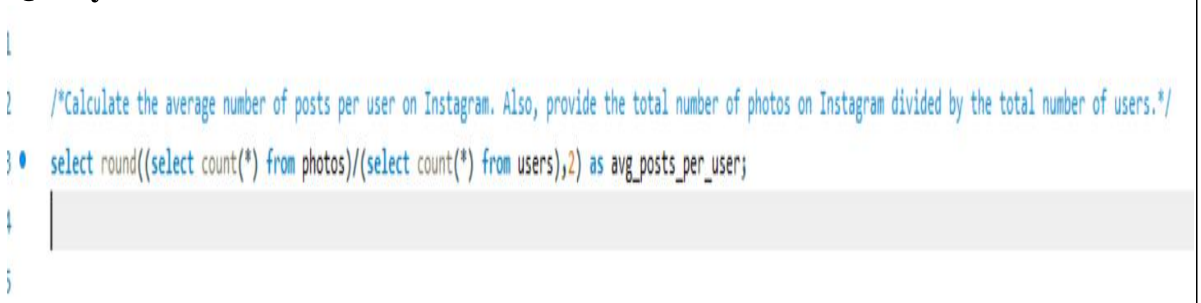
1. **User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts. **Task**: Calculate the average number of posts per user on Instagram. Also,

provide the total number of photos on Instagram divided by the total number of users.

➔ The average number of posts per user on Instagram is achieved by dividing the total number of posts by the total number of users on Instagram.

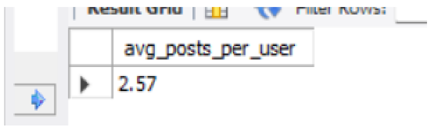
➔ Directly in the select statement, we have divided the total count of photos from photos table by the total count of users from users table, where we got the average no.of posts per user, rounded it off to 2 decimals and named(alias) as avg_posts_per_user.

Query:



```
/*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.*/
select round((select count(*) from photos)/(select count(*) from users),2) as avg_posts_per_user;
```

Output:

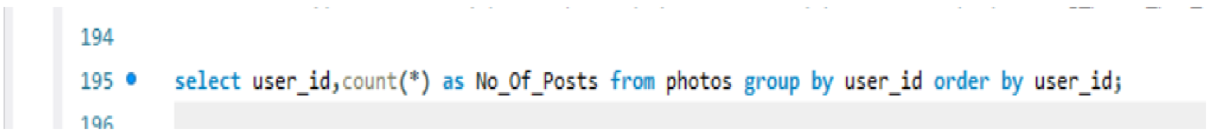


```
avg_posts_per_user
2.57
```

➔ From the output, we got **the average no.of posts per user as 2.57** i.e., total no.of photos =257 and total no.of users=100.

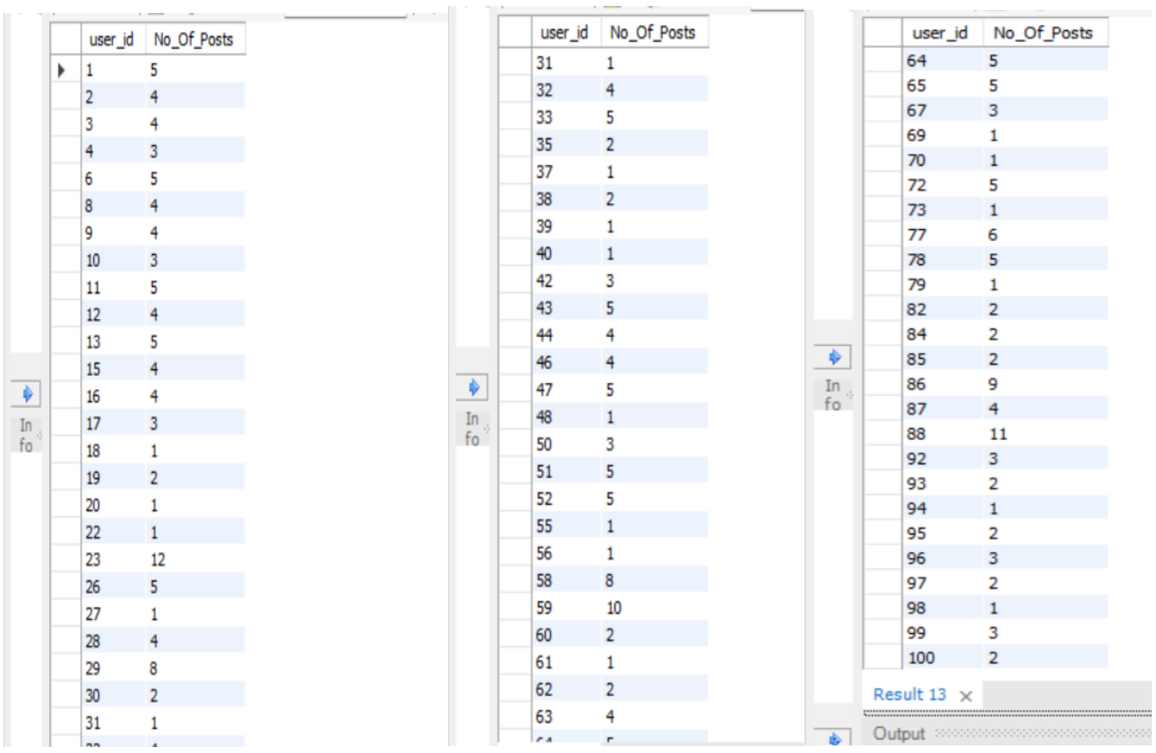
✓ <u>*We can also find the no.of photos posted by each user using the query given below:*</u>

<u>Query:</u>



```
select user_id,count(*) as No_Of_Posts from photos group by user_id order by user_id;
```

<u>Output:</u>

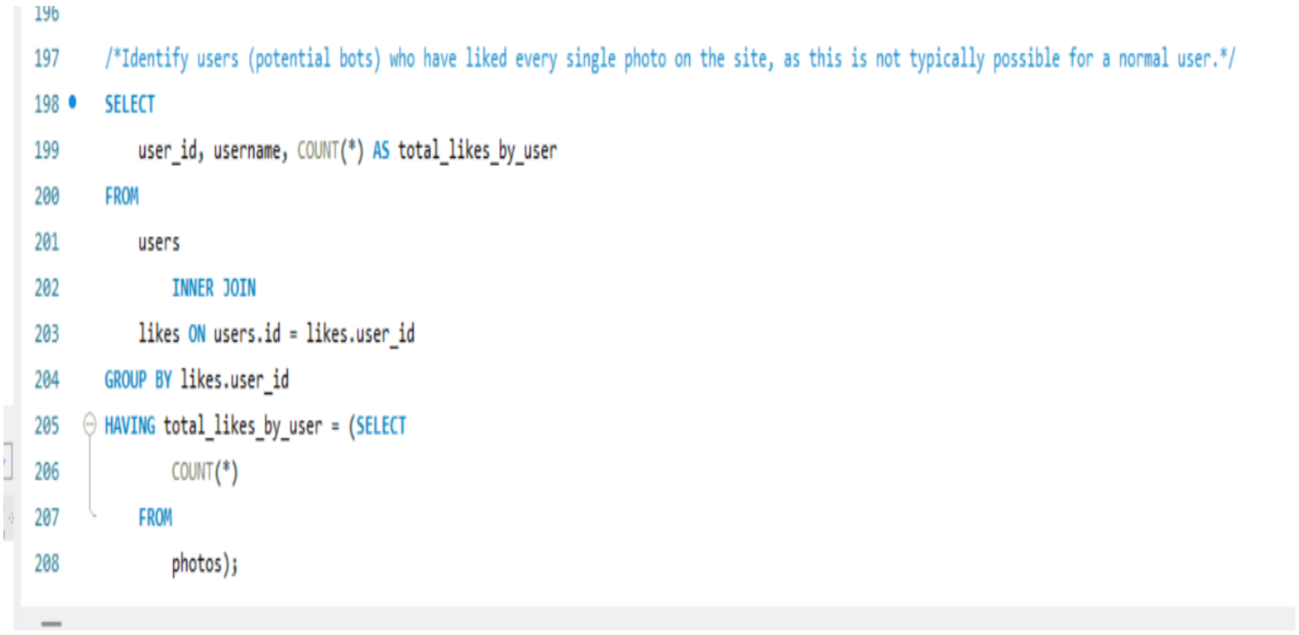


user_id	No_Of_Posts
1	5
2	4
3	4
4	3
6	5
8	4
9	4
10	3
11	5
12	4
13	5
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	12
26	5
27	1
28	4
29	8
30	2
31	1
31	1
32	4
33	5
35	2
37	1
38	2
39	1
40	1
42	3
43	5
44	4
46	4
47	5
48	1
50	3
51	5
52	5
55	1
56	1
58	8
59	10
60	2
61	1
62	2
63	4
64	5
65	5
67	3
69	1
70	1
72	5
73	1
77	6
78	5
79	1
82	2
84	2
85	2
86	9
87	4
88	11
92	3
93	2
94	1
95	2
96	3
97	2
98	1
99	3
100	2

<u>2.Bots & Fake Accounts:</u> Investors want to know if the platform is crowded with fake and dummy accounts.
Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

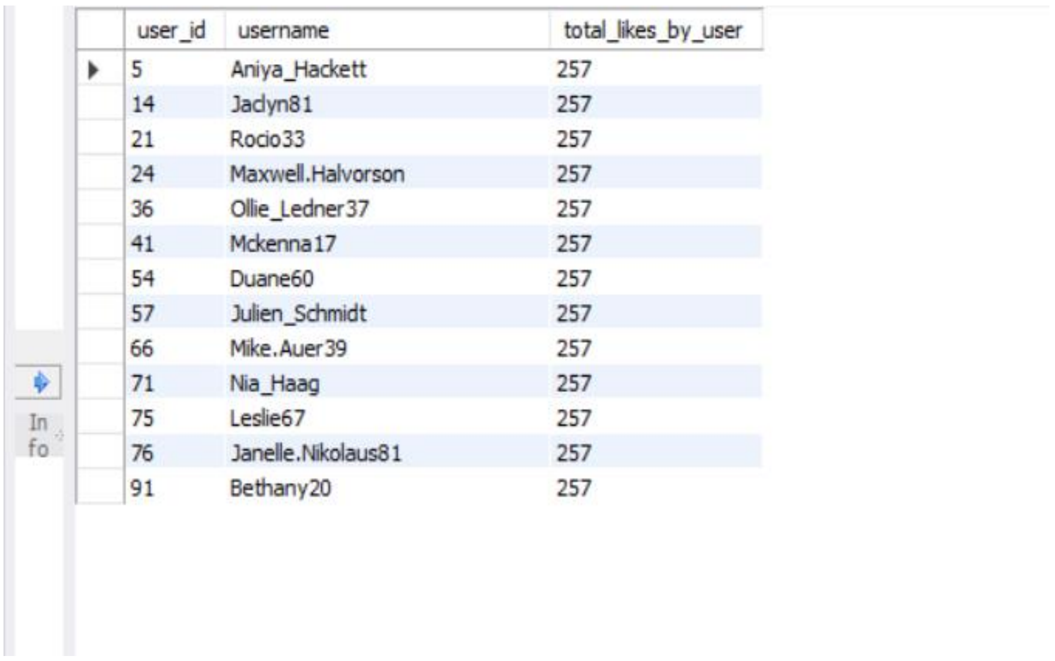
- ➔ Using select statement i.e., SELECT user_id, username, count(*) AS total_likes_by_user FROM users,we select the user_id and username from the users table and counts the total number of likes each user has made and this count is named(alias) as total_likes_by_user
- ➔ Combine the relevant tables users and likes by using JOIN function based on common column from both tables i.e., users.id and likes.user_id. With the help of this join,previously the count(*) function is able to count the total number of likes each user has made.
- ➔ By using the group by function we group the desired output table based on likes.user_id
- ➔ By using 'HAVING total_likes_by_user = (select count(*) from photos', we filtered the results that include only those users whose total_likes_by_user is equal to the total number of photos on Instagram where the subquery (SELECT count(*) FROM photos) calculates the total number of photos.

Query:



```
/*Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.*/
SELECT
    user_id, username, COUNT(*) AS total_likes_by_user
FROM
    users
        INNER JOIN
    likes ON users.id = likes.user_id
GROUP BY likes.user_id
HAVING total_likes_by_user = (SELECT
        COUNT(*)
    FROM
        photos);
```

Output:



user_id	username	total_likes_by_user
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

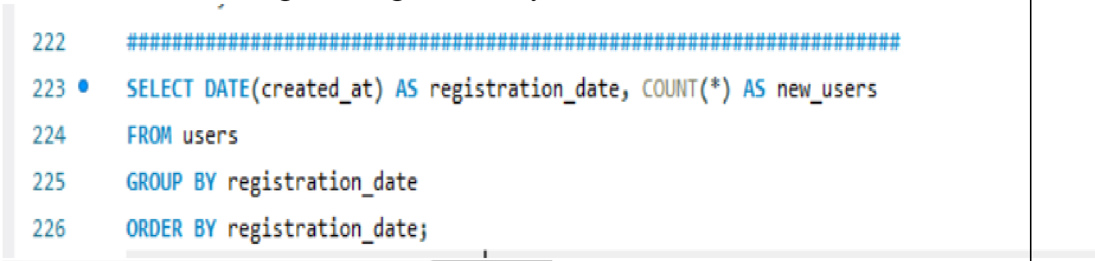
➔ From the result, we got the user_id and username who are likely to be fake accounts or bots since these accounts liked every photo in Instagram(Total number of photos in Instagram=257)

Insights:

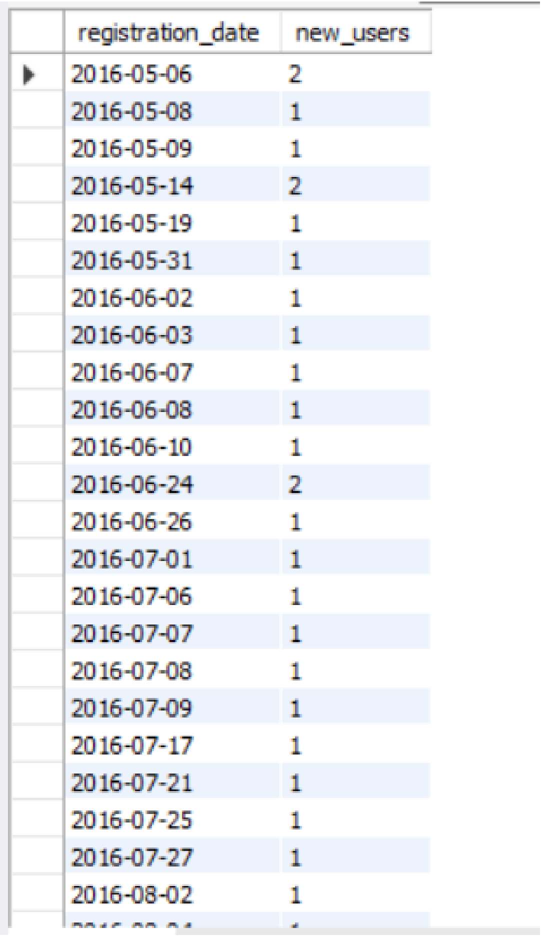
➔ From doing the project, we went through : Identifying the five oldest users on Instagram, Identifying users who have never posted a single photo on Instagram, Identifying the user with the most likes on a single photo, Identifying the top five most commonly used hashtags on the platform, Determining the day of the week when most users register on Instagram, Calculating the average number of photos per user on Instagram, Identifying users(potential bots) who have liked every single photo on the site.

➔ Additionally we can find many more insights like the above tasks in the project. Few of them I found out are :

1. <u>Tracking user growth over time where the query shows the no.of new users registering each day.</u>

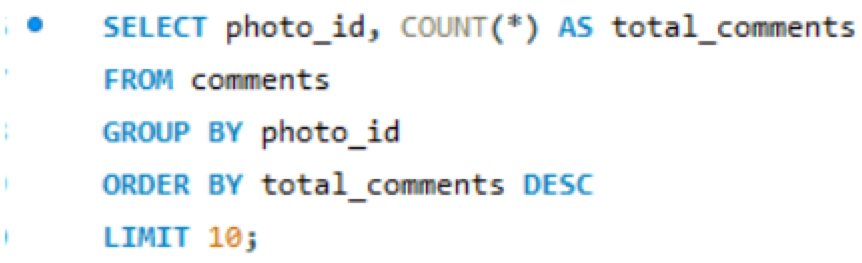


```
####################################################################
SELECT DATE(created_at) AS registration_date, COUNT(*) AS new_users
FROM users
GROUP BY registration_date
ORDER BY registration_date;
```

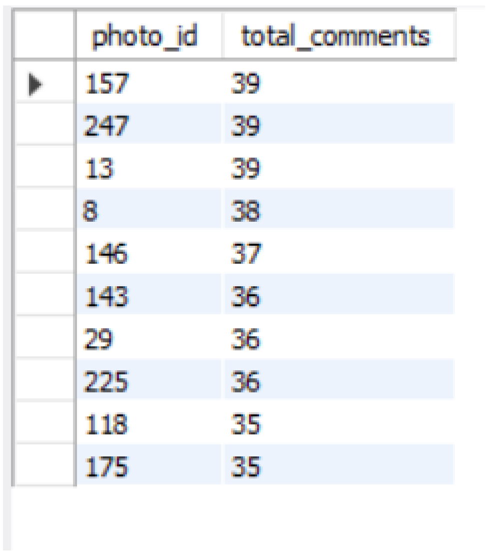


registration_date	new_users
2016-05-06	2
2016-05-08	1
2016-05-09	1
2016-05-14	2
2016-05-19	1
2016-05-31	1
2016-06-02	1
2016-06-03	1
2016-06-07	1
2016-06-08	1
2016-06-10	1
2016-06-24	2
2016-06-26	1
2016-07-01	1
2016-07-06	1
2016-07-07	1
2016-07-08	1
2016-07-09	1
2016-07-17	1
2016-07-21	1
2016-07-25	1
2016-07-27	1
2016-08-02	1

2. <u>Photos with most comments</u>



```
SELECT photo_id, COUNT(*) AS total_comments
FROM comments
GROUP BY photo_id
ORDER BY total_comments DESC
LIMIT 10;
```



photo_id	total_comments
157	39
247	39
13	39
8	38
146	37
143	36
29	36
225	36
118	35
175	35

Result:

Hence,by doing the Instagram User Analytics project, I was able to find out the insights from the ig_clone database. After doing the tasks, I was able to write SQL Queries on my own , enhancing my Data Analytical skills. Using SQL, one can retrieve any kind of information needed from a database and find out potential insights from a raw data(Datbase).