Instagram Users Analytics

Project Description: The project aims to analyze user interactions and engagement within the Instagram app using SQL queries. Its purpose is to provide valuable insights to the management team, guiding strategic decisions for Instagram's future development. We'll focus on extracting insights from the data to understand user behaviour, preferences, and trends.

Approach: We began by understanding the database schema and available datasets, including tables such as users, posts, likes, comments, and followers. Our approach involved formulating SQL queries to extract specific information and address management's questions. We analyzed user activity, engagement metrics, popular content, and demographics to derive meaningful insights.

Tech-Stack Used: We utilized MySQL Workbench for querying the database and analyzing the data. MySQL Workbench offers a user-friendly interface for writing and executing SQL queries, making it suitable for efficient exploration of large datasets. Its robust features, compatibility with MySQL databases, and ease of use were key factors in our choice.

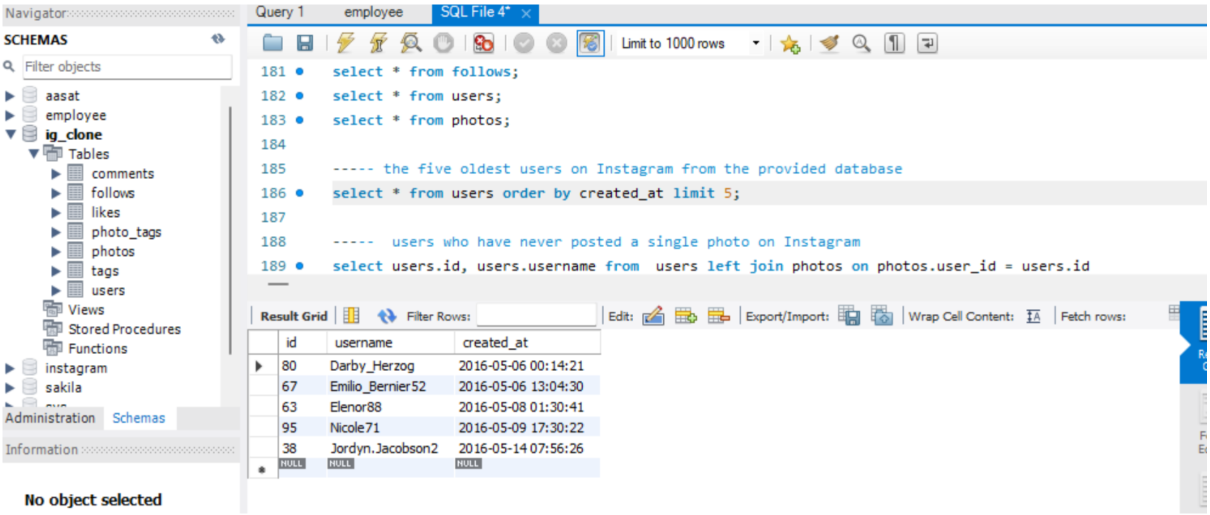
Insights: Our analysis revealed insights into user behaviour and engagement on Instagram. We identified popular posts based on likes and comments, examined user interaction patterns over time, and explored user demographics. Additionally, we uncovered trends in user-generated content and engagement metrics, providing valuable information for decision-making.

Result: The project delivered actionable insights from Instagram user data, empowering the management team to make informed decisions about the app's future. By understanding user interactions and engagement, we identified opportunities to improve user experience, content discovery, and overall engagement on the platform. The analysis contributed to a deeper understanding of Instagram's user base and their preferences, guiding strategic decisions for product development and growth.

A) Marketing Analysis:

Following are the SQL queries used to find out the required data helping the management to make decisions about the app.

1. Executed the sql statement to get the oldest 5 users from the given Database.
   select * from users order by created_at limit 5;



```
select * from follows;
select * from users;
select * from photos;

----- the five oldest users on Instagram from the provided database
select * from users order by created_at limit 5;

-----  users who have never posted a single photo on Instagram
select users.id, users.username from  users left join photos on photos.user_id = users.id
```

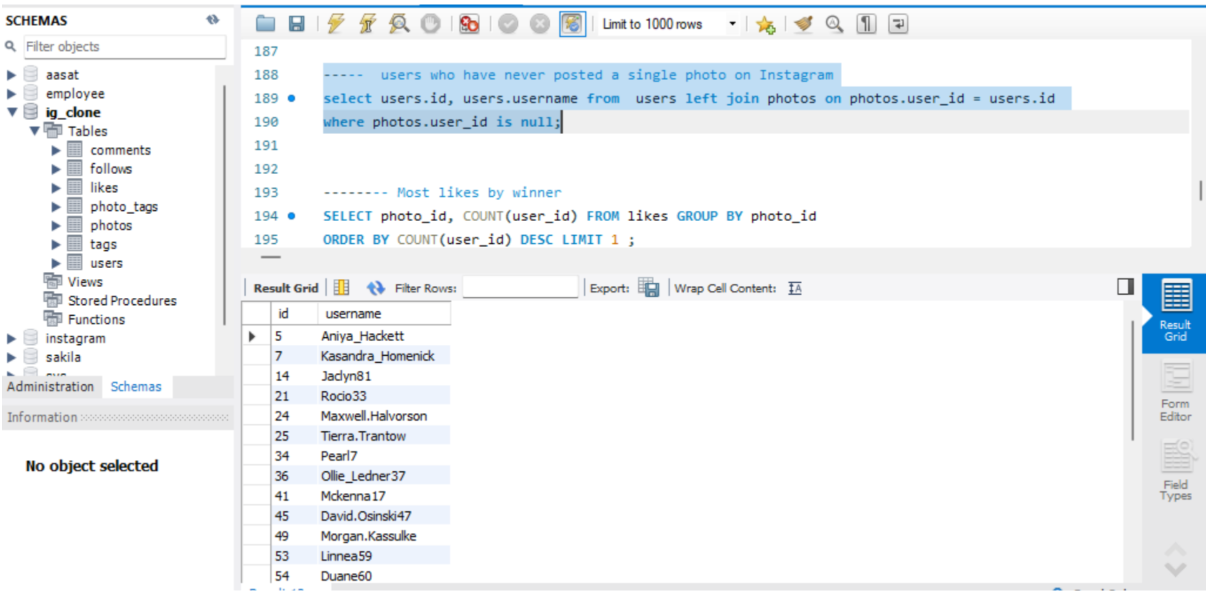
id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

---

2. Executed the Sql statement, to find users, who have never posted a single photo on Instagram. This will help the management to share them promotional E-mails

select users.id, users.username from users left join photos on photos.user_id = users.id

where photos.user_id is null;

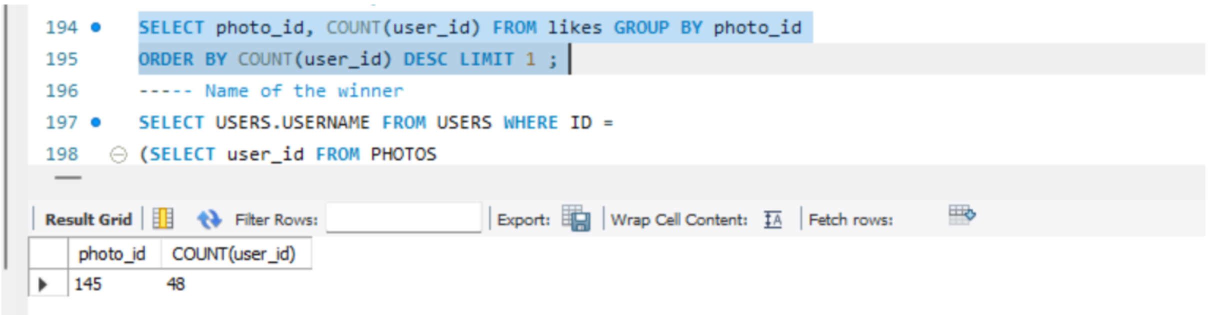


```
-----  users who have never posted a single photo on Instagram
select users.id, users.username from  users left join photos on photos.user_id = users.id
where photos.user_id is null;


-------- Most likes by winner
SELECT photo_id, COUNT(user_id) FROM likes GROUP BY photo_id
ORDER BY COUNT(user_id) DESC LIMIT 1 ;
```

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

3. The team has organized a contest where the user with the most likes on a single photo wins. Sql statement executed to find the Most lokes by winner

SELECT photo_id, COUNT(user_id) FROM likes GROUP BY photo_id ORDER BY COUNT(user_id) DESC LIMIT 1 ;



```
SELECT photo_id, COUNT(user_id) FROM likes GROUP BY photo_id
ORDER BY COUNT(user_id) DESC LIMIT 1 ;
----- Name of the winner
SELECT USERS.USERNAME FROM USERS WHERE ID =
(SELECT user_id FROM PHOTOS
```

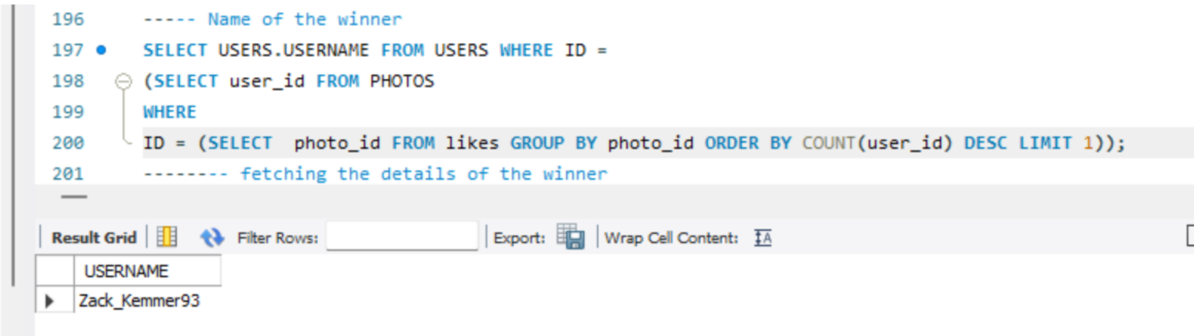
```
photo_id   COUNT(user_id)
145        48
```

Name of the winner

SELECT USERS.USERNAME FROM USERS WHERE ID =

(SELECT user_id FROM PHOTOS WHERE

ID = (SELECT photo_id FROM likes GROUP BY photo_id ORDER BY COUNT(user_id) DESC LIMIT 1));



```
----- Name of the winner
SELECT USERS.USERNAME FROM USERS WHERE ID =
(SELECT user_id FROM PHOTOS
WHERE
ID = (SELECT  photo_id FROM likes GROUP BY photo_id ORDER BY COUNT(user_id) DESC LIMIT 1));
-------- fetching the details of the winner
```

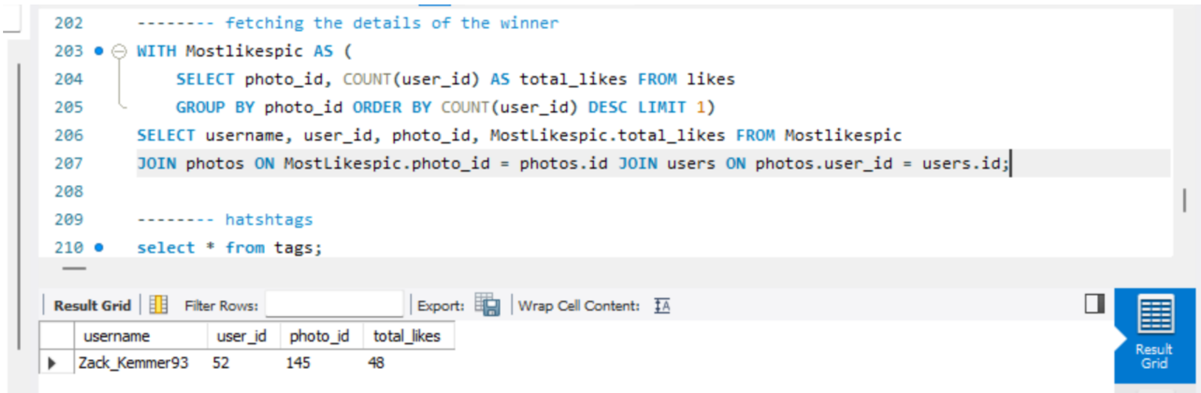
```
USERNAME
Zack_Kemmer93
```

To find the details of the winner following is the SQL statement used.

WITH Mostlikespic AS (SELECT photo_id, COUNT(user_id) AS total_likes FROM likes

GROUP BY photo_id ORDER BY COUNT(user_id) DESC LIMIT 1) SELECT username, user_id, photo_id, MostLikespic.total_likes FROM Mostlikespic

JOIN photos ON MostLikespic.photo_id = photos.id JOIN users ON photos.user_id = users.id;

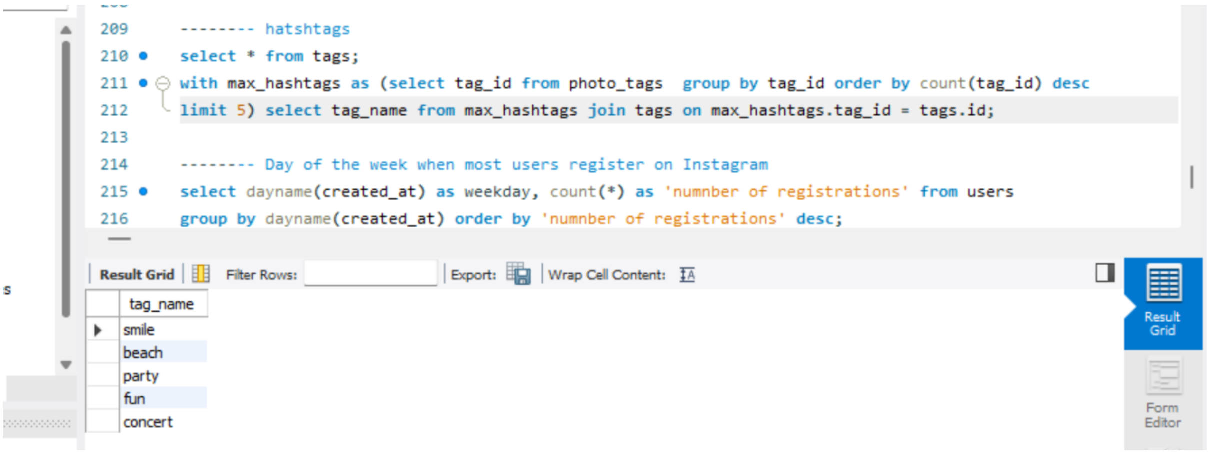


```
-------- fetching the details of the winner
WITH Mostlikespic AS (
    SELECT photo_id, COUNT(user_id) AS total_likes FROM likes
    GROUP BY photo_id ORDER BY COUNT(user_id) DESC LIMIT 1)
SELECT username, user_id, photo_id, MostLikespic.total_likes FROM Mostlikespic
JOIN photos ON MostLikespic.photo_id = photos.id JOIN users ON photos.user_id = users.id;

-------- hatshtags
select * from tags;
```

```
username        user_id   photo_id   total_likes
Zack_Kemmer93   52        145        48
```

4. Identifying and suggesting the top five most commonly used hashtags on the platform. Following is the SQL statement used to help the partner brand to know most popular hashtags.

   select * from tags;
   with max_hashtags as (select tag_id from photo_tags group by tag_id order by count(tag_id) desc limit 5) select tag_name from max_hashtags join tags on max_hashtags.tag_id = tags.id;



```
-------- hatshtags
select * from tags;
with max_hashtags as (select tag_id from photo_tags  group by tag_id order by count(tag_id) desc
limit 5) select tag_name from max_hashtags join tags on max_hashtags.tag_id = tags.id;

-------- Day of the week when most users register on Instagram
select dayname(created_at) as weekday, count(*) as 'numnber of registrations' from users
group by dayname(created_at) order by 'numnber of registrations' desc;
```

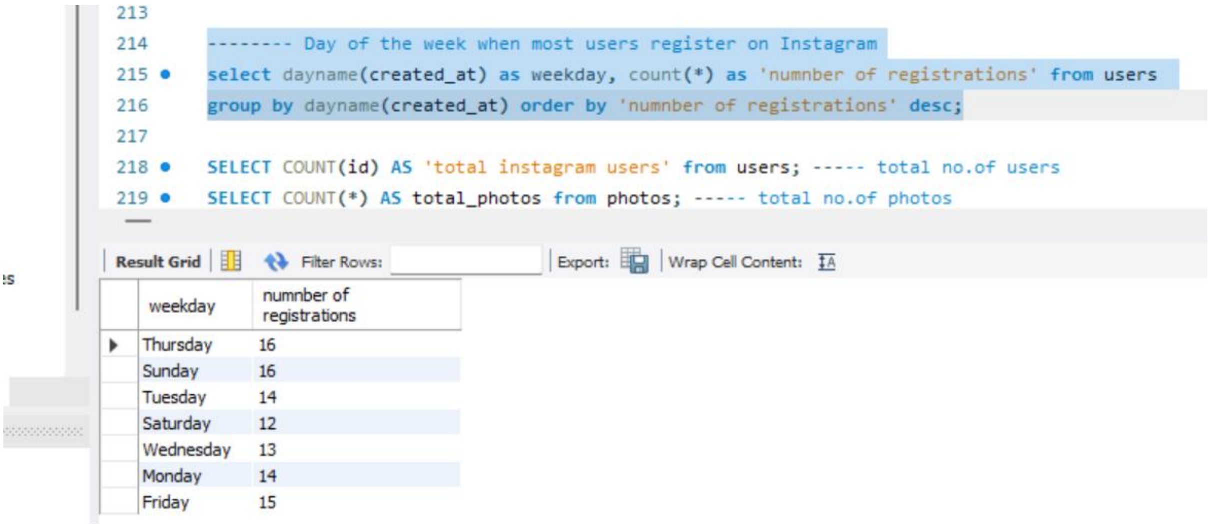
tag_name
smile
beach
party
fun
concert

---

5. The team wants to know the best day of the week to launch ads. Following is the SQl statement used to find out the Day of the week when most of the users register on Instagram. As per the outcome **Thursdays and Sundays** are the best days to promote campaign followed by **Tuesdays.**

select dayname(created_at) as weekday, count(*) as 'numnber of registrations' from users

group by dayname(created_at) order by 'numnber of registrations' desc;



```

-------- Day of the week when most users register on Instagram
select dayname(created_at) as weekday, count(*) as 'numnber of registrations' from users
group by dayname(created_at) order by 'numnber of registrations' desc;

SELECT COUNT(id) AS 'total instagram users' from users; ----- total no.of users
SELECT COUNT(*) AS total_photos from photos; ----- total no.of photos
```

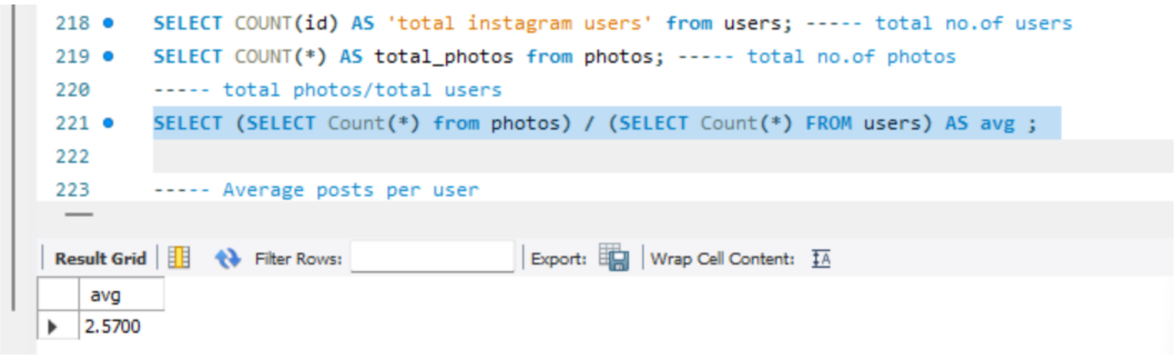
weekday	numnber of registrations
Thursday	16
Sunday	16
Tuesday	14
Saturday	12
Wednesday	13
Monday	14
Friday	15

B) Investor Metrics:

1. Investors want to know if users are still active and posting on Instagram or if they are making fewer posts. Following are the SQL queries to find the requirements.

   SELECT COUNT(id) AS 'total instagram users' from users;   ----- total no.of users
   SELECT COUNT(*) AS total_photos from photos;   ----- total no.of photos

   ----- total photos/total users
   SELECT (SELECT Count(*) from photos) / (SELECT Count(*) FROM users) AS avg ;



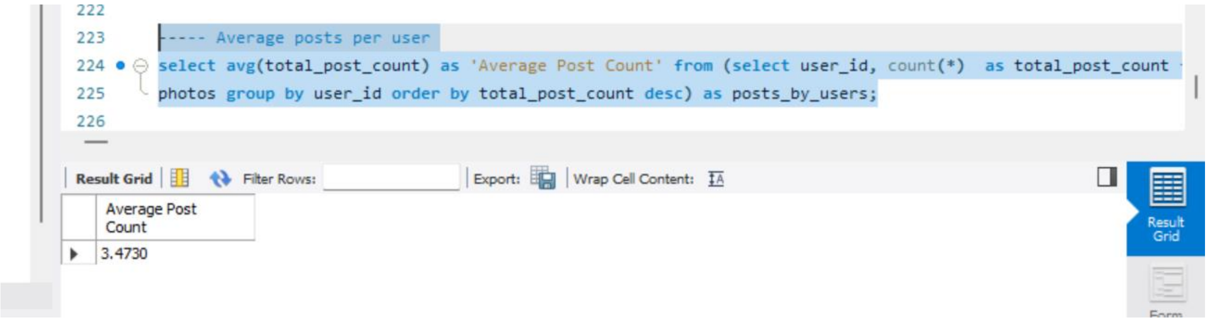
```
SELECT COUNT(id) AS 'total instagram users' from users; ----- total no.of users
SELECT COUNT(*) AS total_photos from photos; ----- total no.of photos
----- total photos/total users
SELECT (SELECT Count(*) from photos) / (SELECT Count(*) FROM users) AS avg ;

----- Average posts per user
```

avg
2.5700

So total Photos per instagram users are 257/100=2.57

The average number of posts per user on Instagram is found using the following SQL query.

select avg(total_post_count) as 'Average Post Count' from (select user_id, count(*) as total_post_count from
photos group by user_id order by total_post_count desc) as posts_by_users;



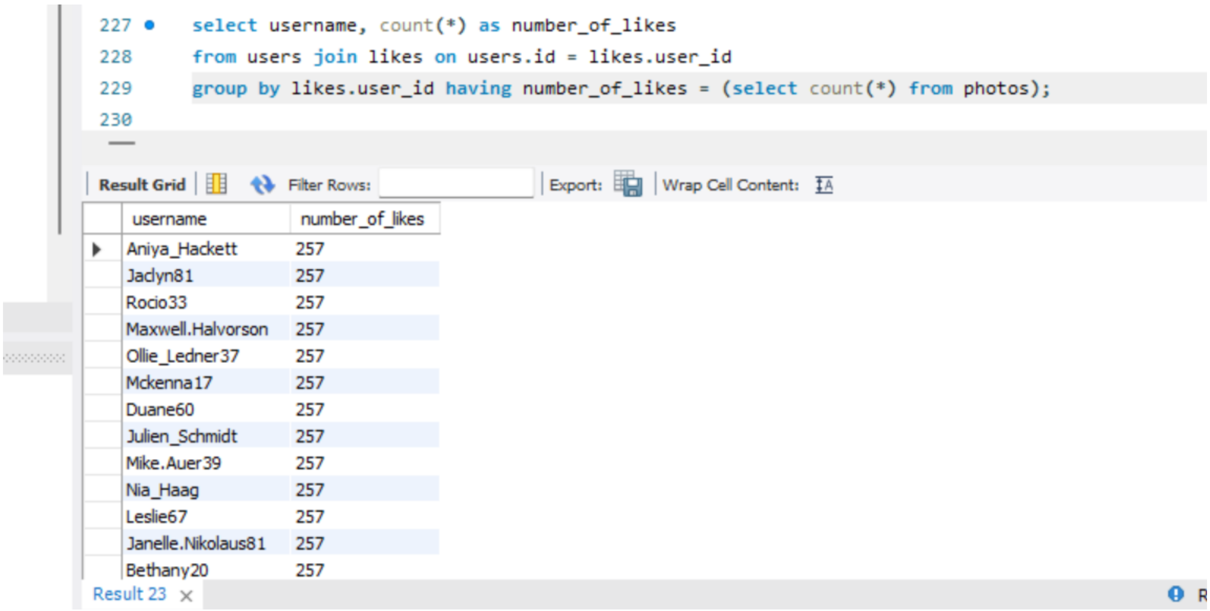
```

----- Average posts per user
select avg(total_post_count) as 'Average Post Count' from (select user_id, count(*)  as total_post_count
photos group by user_id order by total_post_count desc) as posts_by_users;

```

Average Post Count
3.4730

2. Investors want to know if the platform is crowded with fake and dummy accounts. Following is the SQL executed to find out Bots or Dummy/Fake accounts.

select username, count(*) as number_of_likes from users join likes on users.id = likes.user_id

group by likes.user_id having number_of_likes = (select count(*) from photos);



```
select username, count(*) as number_of_likes
from users join likes on users.id = likes.user_id
group by likes.user_id having number_of_likes = (select count(*) from photos);
```

username	number_of_likes
Aniya_Hackett	257
Jaclyn81	257
Rocio33	257
Maxwell.Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bethany20	257

From the above provided SQL queries the Management should get the insights or help onto when to promote campaigns, offers, choosing winners and promotional mails to inactive users.

Thank You.