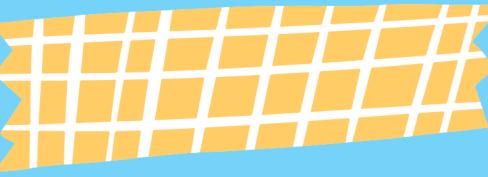




INSTAGRAM USER ANALYTICS



INTRODUCTION

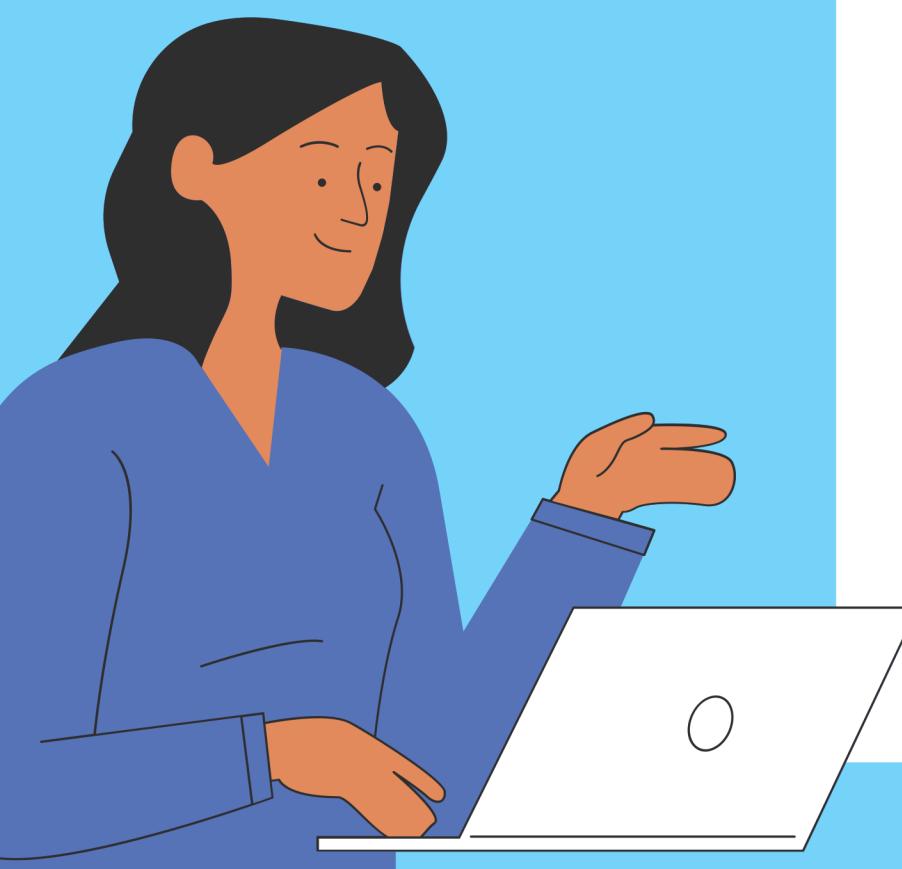
HARSH RAJ

BUSINESS SYSTEM ANALYST

@ AXIS BANK

DATA ANALYTICS ENTHUSIAST





Give a brief about your project description i.e. what is this project about, how are you going to handle the things and what are the things that you are going to find out through the project.

DESCRIPTION



This project is focused on analyzing user engagement and provide insights to marketing and product teams.

In this project we are going to cover various topics related to marketing campaigns, business evaluation and customer engagement by analyzing the user data.

The analysis will include identifying the oldest users, inactive users, the most liked photo, the most used hashtags, and the best day to post an ad.

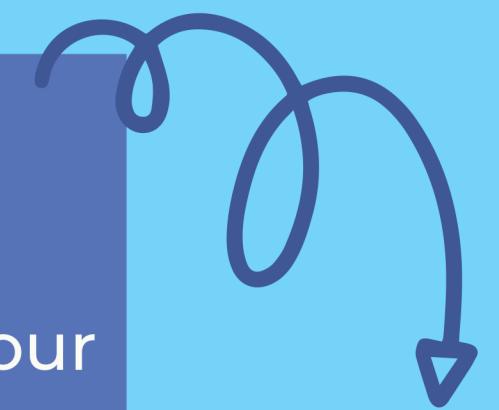
Also, we will monitor user engagement and fake accounts.

This project will involve creation of database and analyzing the data to derive insights.



APPROACH

Write a short paragraph about your approach towards the project and how you have executed it.



To complete this project, first we will download and install **MySQL** and we will use **SQL** queries to extract the required information from the provided database. By use different **SQL** commands and data manipulation techniques, we will perform the analysis. After that it will be presented in a report format.



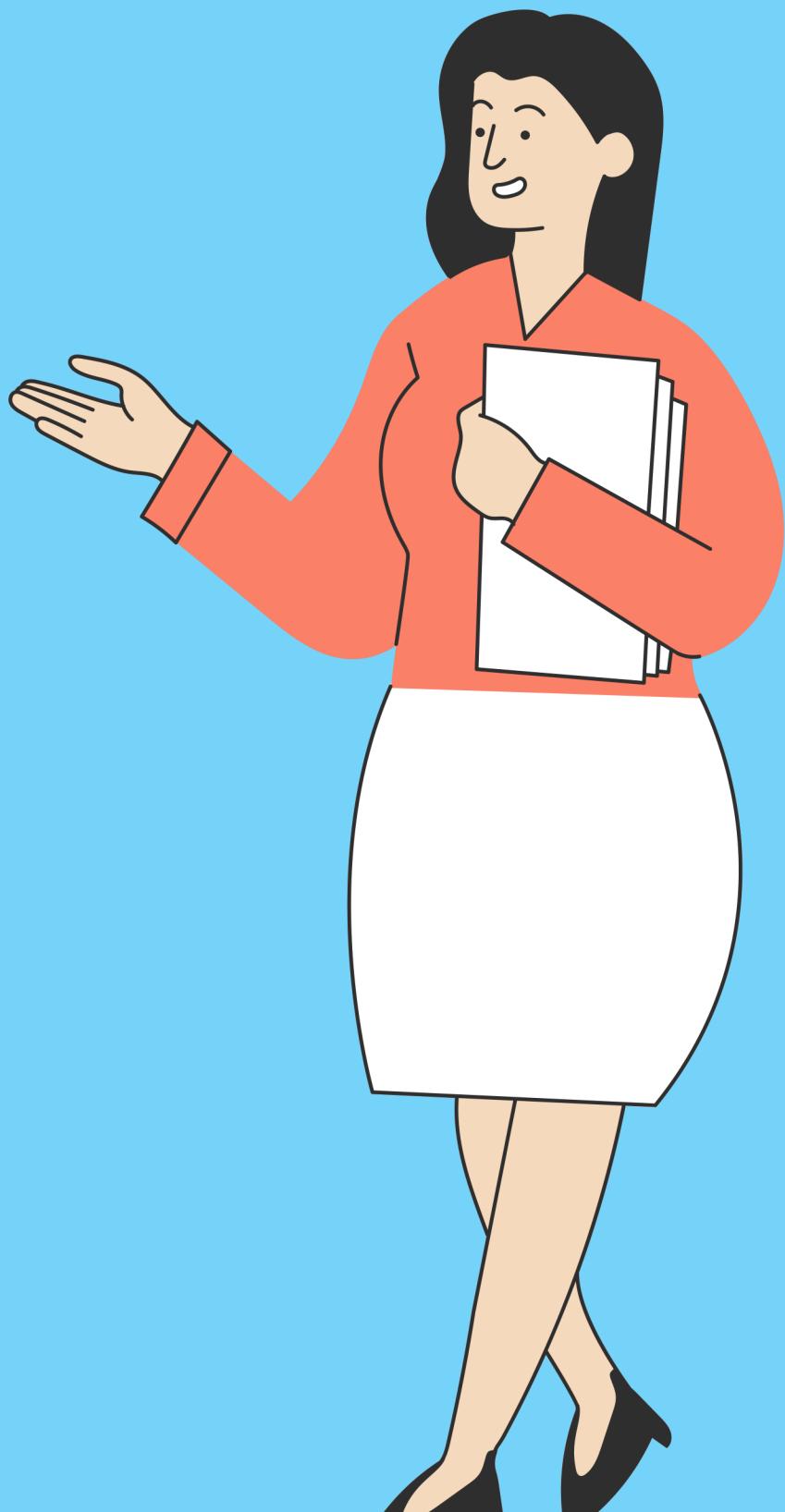


TECH-STACK USED

Do mention the software and the version used while making the project and mention the purpose of using it.

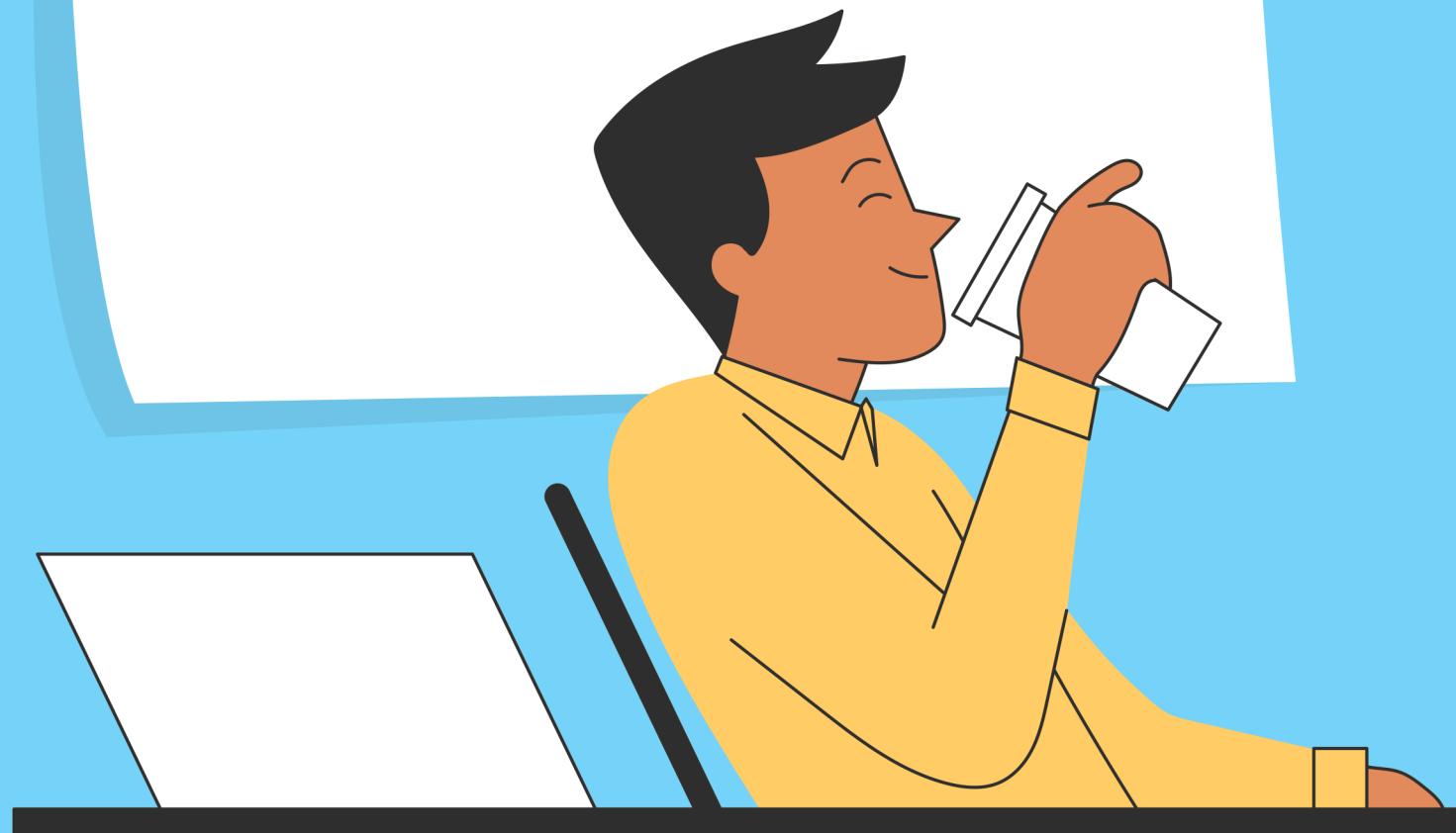


For this project, we will use **SQL** to perform the analysis and the **SQL** queries were executed using a database management system i.e. **MySQL**.



AGENDA

MARKETING:



The Marketing team wants to launch some campaigns, and they need your help with the following.....

INSIGHTS & RESULTS:

Rewarding Most Loyal Users:

People who have been using the platform for the longest time.

Your Task: Find the 5 oldest users of the Instagram from the database provided

a) To find the Most loyal Users / Oldest Users we will use the data from the user table and select **username** and **created_at** column, So it will give the whole record but we need top 5 and in ascending order.

b) Now we will use **Order By** function to order it and apply it with **created_at** column in **ascending** order.

c) Now we will use the **Limit** and the output will be displayed for top 5 oldest users.

```
mysql> select username,created_at from users order by created_at ASC limit 5;
+-----+-----+
| username | created_at |
+-----+-----+
| Darby_Herzog | 2016-05-06 00:14:21 |
| Emilio_Bernier52 | 2016-05-06 13:04:30 |
| Elenor88 | 2016-05-08 01:30:41 |
| Nicole71 | 2016-05-09 17:30:22 |
| Jordyn.Jacobson2 | 2016-05-14 07:56:26 |
+-----+
5 rows in set (0.00 sec)
```

INSIGHTS & RESULTS:

Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.

Your Task: Find the users who have never posted a single photo on Instagram

a) By analyzing the database for users who never posted a photo on Instagram, we can identify the users who are inactive.

b) So we will select **username** column from the users table and then we will join photos table with user table on **users.id=photos.user_id**

c) After that we will also find the rows from the users table where **photo.id is NULL**.

So Final Query will be:

```
SELECT USERS.USERNAME,USERS.ID FROM USERS LEFT JOIN PHOTOS ON  
USERS.ID=PHOTOS.USER_ID WHERE PHOTOS.ID IS NULL;
```

```
mysql> select users.username,users.id from users left join photos on users.id=photos.user_id where photos.id is NULL;
+-----+---+
| username        | id |
+-----+---+
| Aniya_Hackett   | 5  |
| Kasandra_Homenick | 7  |
| Jaclyn81        | 14 |
| Rocio33          | 21 |
| Maxwell.Halvorson | 24 |
| Tierra.Trantow   | 25 |
| Pearl17          | 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 |
+-----+---+
26 rows in set (0.00 sec)
```

INSIGHTS & RESULTS:

Declaring Contest Winner: The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.

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

- a) This analysis is to identify the user who received the most likes on a single photo, and he/she will be winner
- b) So first we will select **users.username ,photos.id, photos.image_url and count(*) as total_likes**.
- c) Now we will join all the three table i.e. **users,photos, likes on photos.id=likes.photo_id ,users.id=likes.photo_id**
- d) Now, we will use group by and we will group the output on the basis of **photos.id**
- e) Then using order by function we will sort in descending order but just because it will show all the data so we will use Limit function to view only the top one.

```
mysql> select users.username,photos.id,photos.image_url,count(*) as total_likes
-> from likes join photos on photos.id=likes.photo_id join users on users.id=likes.photo_id
-> group by photos.id order by total_likes desc limit 1;
+-----+----+-----+-----+
| username | id | image_url      | total_likes |
+-----+----+-----+-----+
| Kaley9   | 30 | http://kenny.com |          41 |
+-----+----+-----+-----+
1 row in set (0.01 sec)
```

INSIGHTS & RESULTS:

Hashtag Researching: A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.

Your Task: Identify and suggest the top 5 most commonly used hashtags on the platform

a) We will analyze the frequency of hashtag usage in the database, from that we can determine the top 5 most commonly used hashtags.

b) So first we will select **tag_name** column from tag table and **count(tag_name)** to count number of tags used.

c) Then we will join **photo_tags** table and tags table on **tags.id =photo_tags.tag_id**

d) After that we will use group by function on the basis of **tags.id**

e) then we will use **Order By** function to sort in descending order on the basis of total

f) Then to find top 5 tag which is used we will use **Limit 5**

```
mysql> SELECT tag_name, COUNT(tag_name) AS total FROM tags JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tags.id ORDER BY total DESC LIMIT 5;
+-----+-----+
| tag_name | total |
+-----+-----+
| smile    | 59   |
| beach    | 42   |
| party    | 39   |
| fun      | 38   |
| concert  | 24   |
+-----+-----+
5 rows in set (0.00 sec)
```

Launch AD Campaign: The team wants to know, which day would be the best day to launch ADs.

Your Task: What day of the week do most users register on? Provide insights on when to schedule an ad campaign

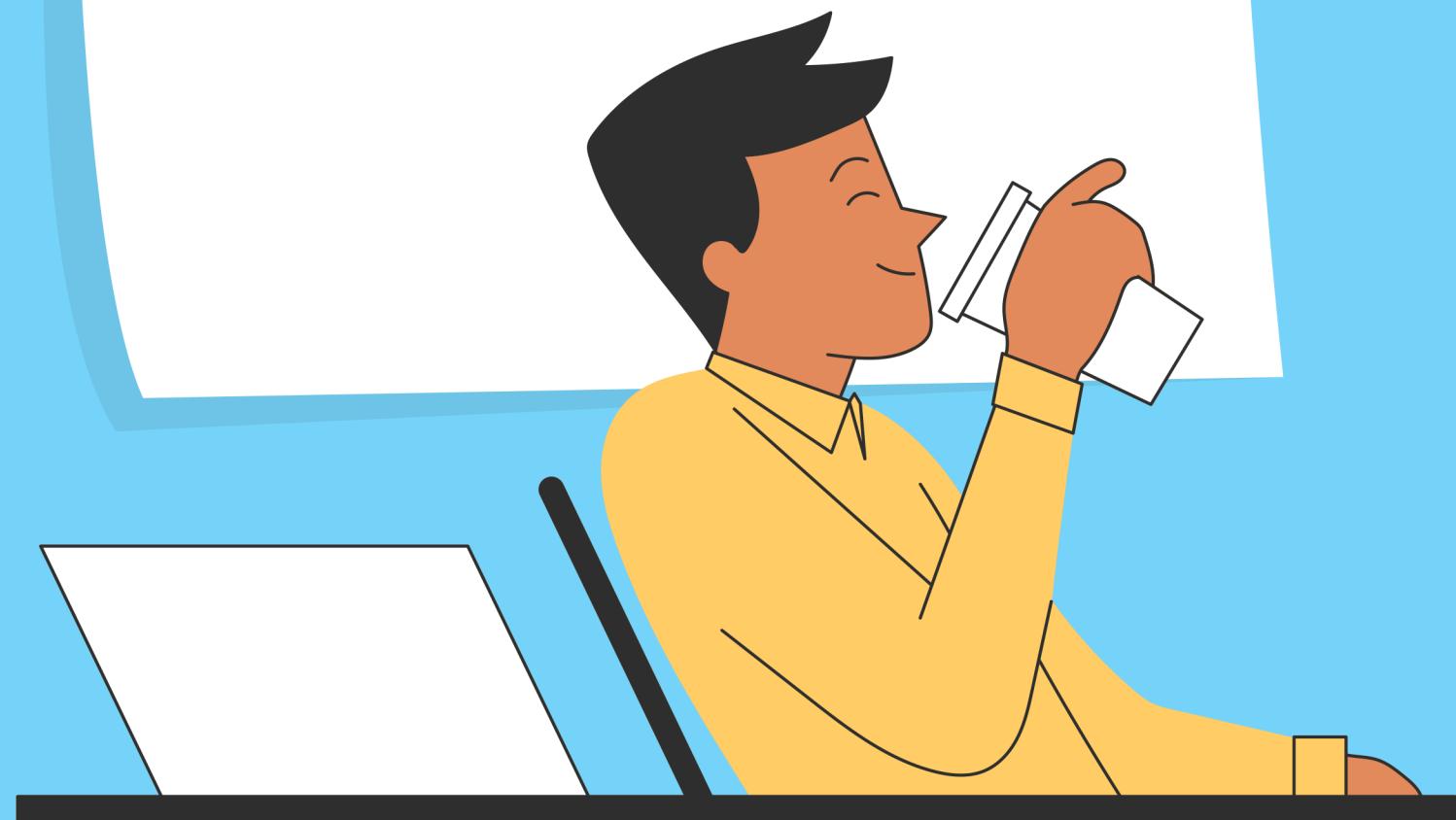
INSIGHTS & RESULTS:

- a) We will analyze the user registration data to identify the day of the week with the highest number of new user registrations. The output will help in scheduling ad campaigns on the best day.
- b) First we will select **date_format as days** and then **Count(*) as No. Of Registration from Users**
- c) After that we will use **Group By** function on the basis of days
- d) In Last we will use **Order By** on the basis of No. of registration in descending order .
- e) Here we can see on Thursday and Sunday Most no of Registration is done.

```
mysql> SELECT date_format(created_at, '%W') AS 'Days', COUNT(*) AS 'No. of Registration' from users group by days order by 2 desc;
+-----+-----+
| Days      | No. of Registration |
+-----+-----+
| Thursday   |          16 |
| Sunday     |          16 |
| Friday     |          15 |
| Tuesday    |          14 |
| Monday     |          14 |
| Wednesday  |          13 |
| Saturday   |          12 |
+-----+-----+
7 rows in set (0.11 sec)
```

AGENDA

INVESTOR METRICS:



Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds.....

INSIGHTS & RESULTS:

User Engagement: Are users still as active and post on Instagram or they are making fewer posts

Your Task: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users

By calculating the average number of posts per user, we can get the level of user engagement. Also, by comparing the total number of photos to the total number of users, we can determine the average number of photos per user.

a)First we will derive average post count by using:

```
SELECT AVG(post_count) AS average_post_count FROM (SELECT user_id, COUNT(*) AS post_count FROM photos GROUP BY user_id) AS subquery;
```

b)Then we will find total photo divided by total user by using:

```
SELECT ROUND((SELECT COUNT(*) FROM photos)/(SELECT COUNT(*) FROM users), 2) AS total_photo_diivide_by_total_users;
```

c)In Last we will find total occurrence of each user Id by using:

```
SELECT user_id, COUNT(*) AS total_occurrence FROM photos GROUP BY user_id;
```

```
mysql> SELECT AVG(post_count) AS average_post_count FROM (SELECT user_id, COUNT(*) AS post_count FROM photos GROUP BY user_id) AS subquery;
+-----+
| average_post_count |
+-----+
|          3.4730 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT ROUND((SELECT COUNT(*) FROM photos)/(SELECT COUNT(*) FROM users), 2) AS total_photo_diivide_by_total_users;
+-----+
| total_photo_diivide_by_total_users |
+-----+
|              2.57 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT user_id, COUNT(*) AS total_occurrence FROM photos GROUP BY user_id;
+-----+-----+
| user_id | total_occurrence |
+-----+-----+
| 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               |
| 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

74 rows in set (0.00 sec)

INSIGHTS & RESULTS:

Bots & Fake Accounts: The investors want to know if the platform is crowded with fake and dummy accounts

Your Task: Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

- a) This analysis is done to identifying users who have liked every single photo on the site, which indicates the presence of potential bots or fake accounts.
- b) First we will select the **user_id**, **username** table from users table and select count(*) function to count **total_likes_by_user** from likes table.
- c) Then we join users and likes table on the basis of **users.id** and **likes.user_id**
- d) Then we use the group by function on the basis of **likes.user_id** for desired output.
- e) In last we will use select command with count(*) from photos having equal values with the **total_likes_by_user**.

```
mysql> SELECT users.id,username, COUNT(users.id) As total_likes_by_user FROM users
    -> JOIN likes ON users.id = likes.user_id GROUP BY users.id HAVING total_likes_by_user = (SELECT COUNT(*) FROM photos);
+----+-----+-----+
| 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 |
+----+-----+-----+
13 rows in set (0.01 sec)
```



THANK YOU!

Have a
great day
ahead.