



INSTAGRAM USER ANALYTICS



BY- SACHIN S

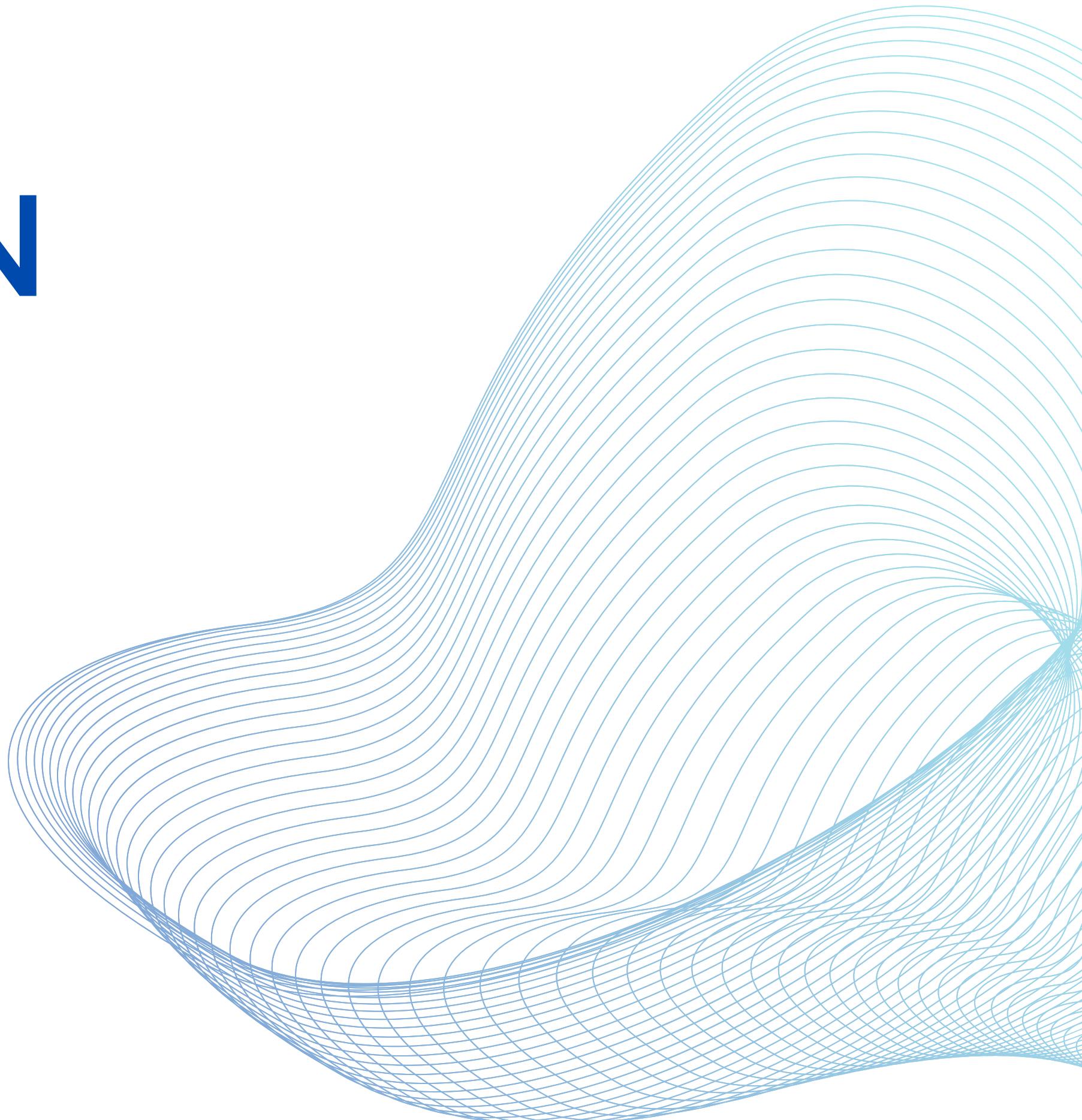
TABLE OF CONTENT

- Introduction
- Description
- Approach
- Tech-Stack Used
- Insights

INTRODUCTION

This project aims to analyze the user interaction and the usage of the Instagram app. We use dataset which contains details of users, their posts, likes and comments on their posts.

Two verticals which are used to analyze user interaction and help investors and marketing team improve their value are **marketing** and **investor metrics**.

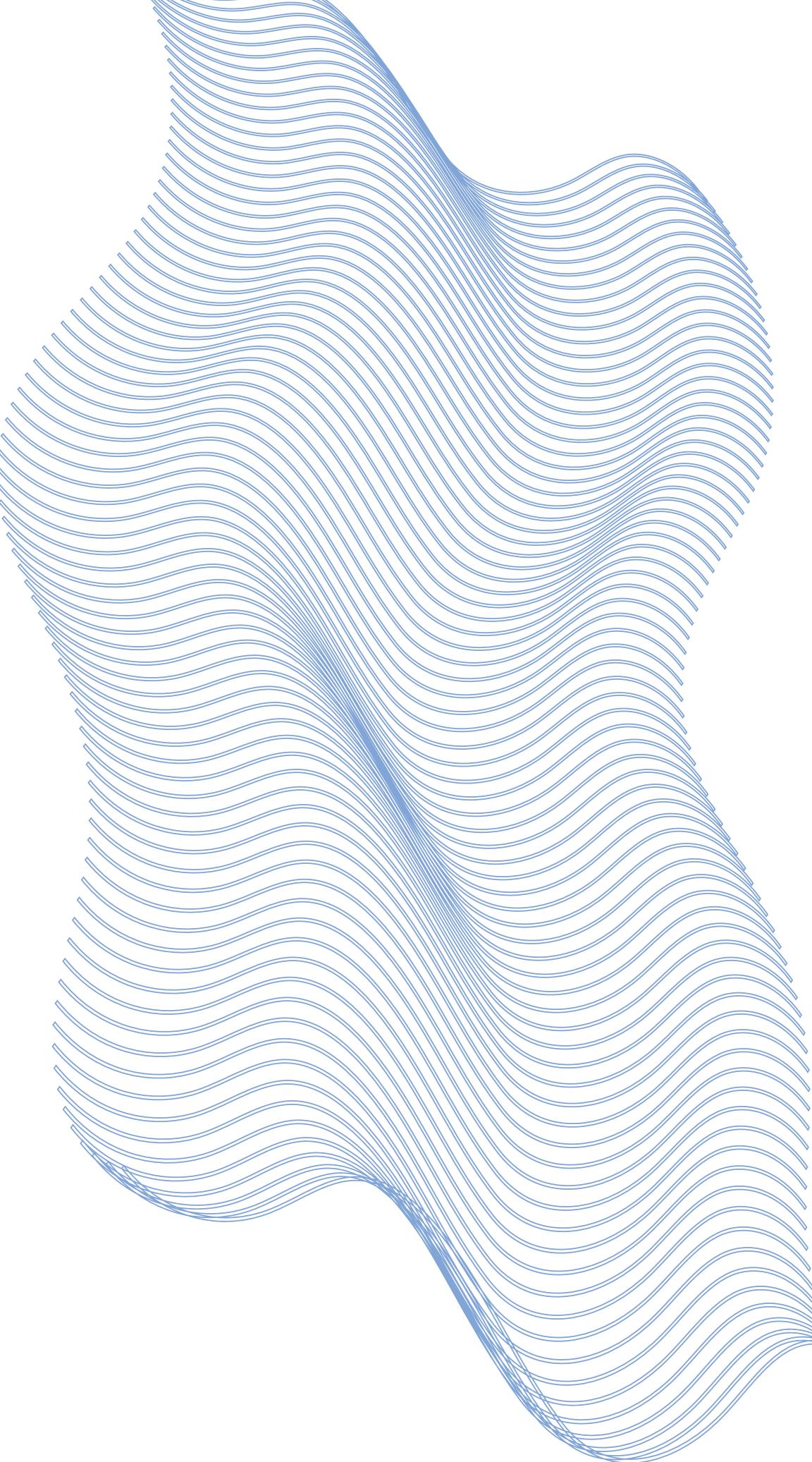


DESCRIPTION

The project has the data of users of the app and helps the team to market the app and increase the value of the company.

There are few categories under which team gives prizes and markets the company to increase it's value. We will be using SQL which gives the selected users from the dataset.

Dataset contains certain tables and values.



APPROACH

The dataset contains details of users, their posts and the likes and comments on their posts.

To market the company there are certain strategies like looking for the most active users and user with most likes on posts etc.

To find the users under certain given criteria we use SQL to get the output from the large dataset.

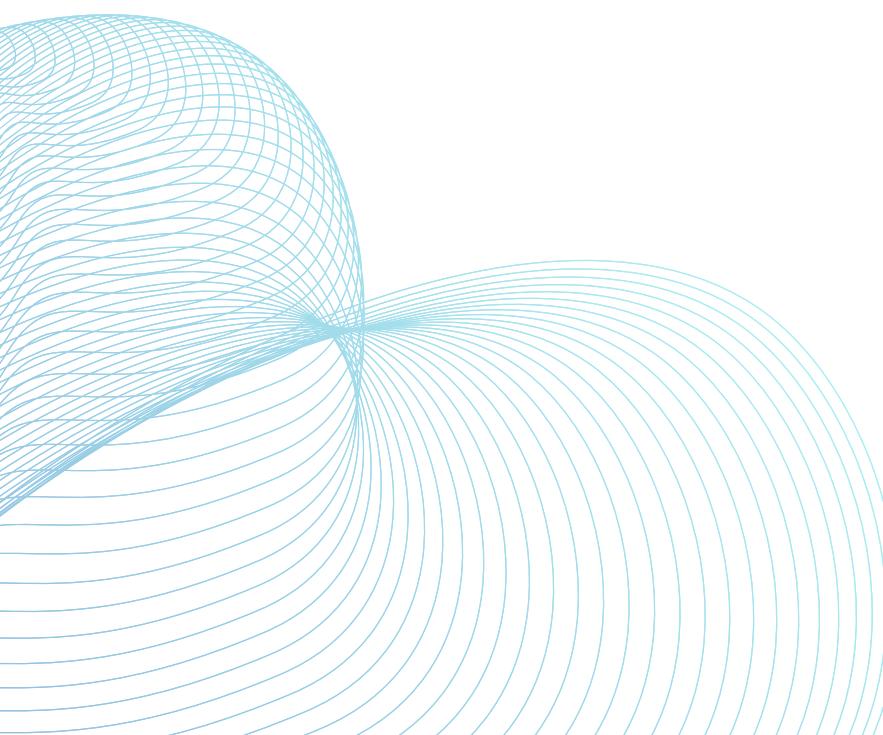
For certain strategies we would have to use **joins** to get the output.

TECH-STACK USED

The tech-stack used to work on this project is

SQL WORKBENCH 8.0 CE (version 8.0.32).

SQL helps to retrieve data from the large dataset.



INSIGHTS

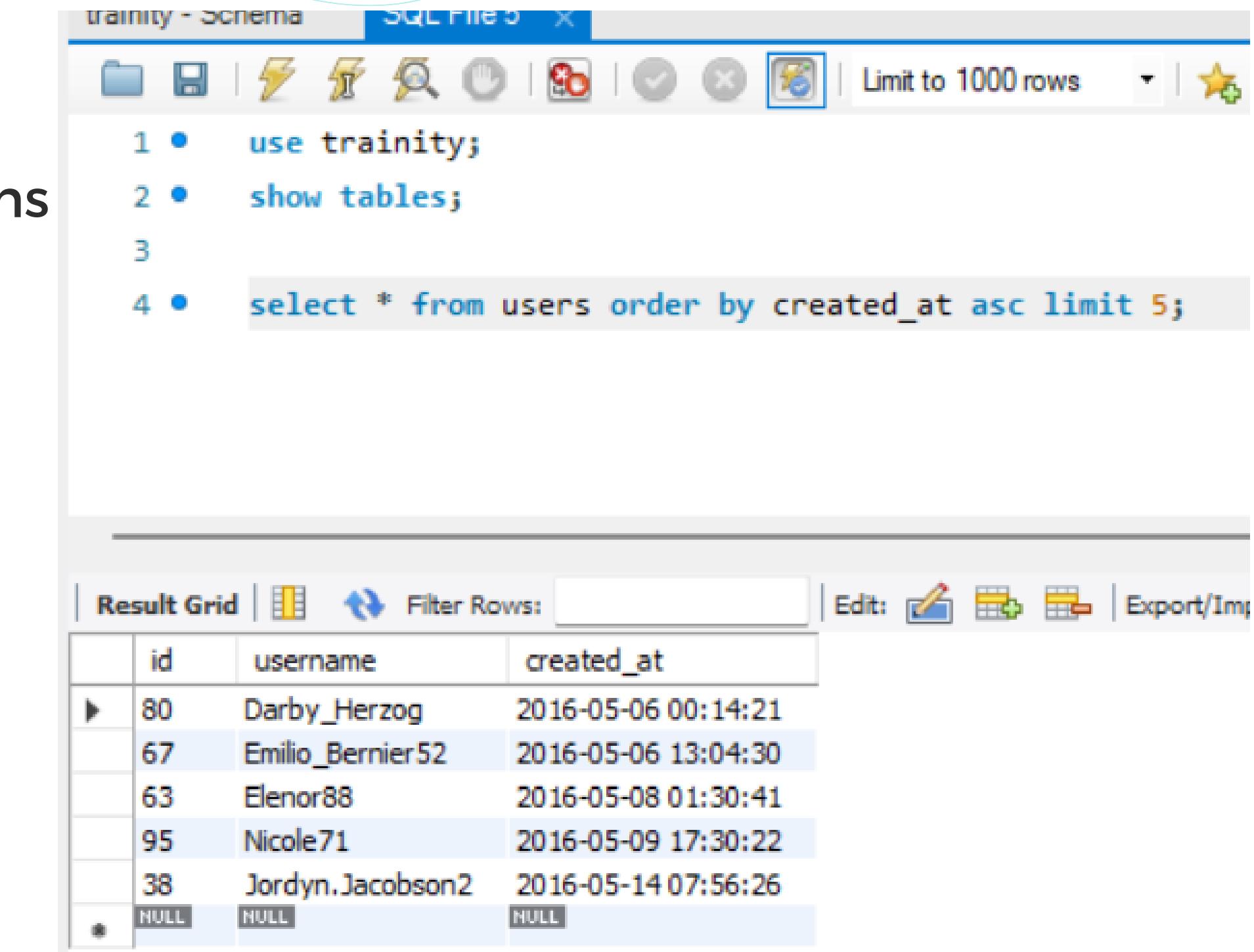
To market the company there are certain criterias and campaigns. Awarding the most active users, awarding the users having most liked posts, awarding users who joined earliest.



1. REWARDING MOST LOYAL USERS

To market the company certain campaigns are held to reward the most loyal users.

People using the app for the longest time



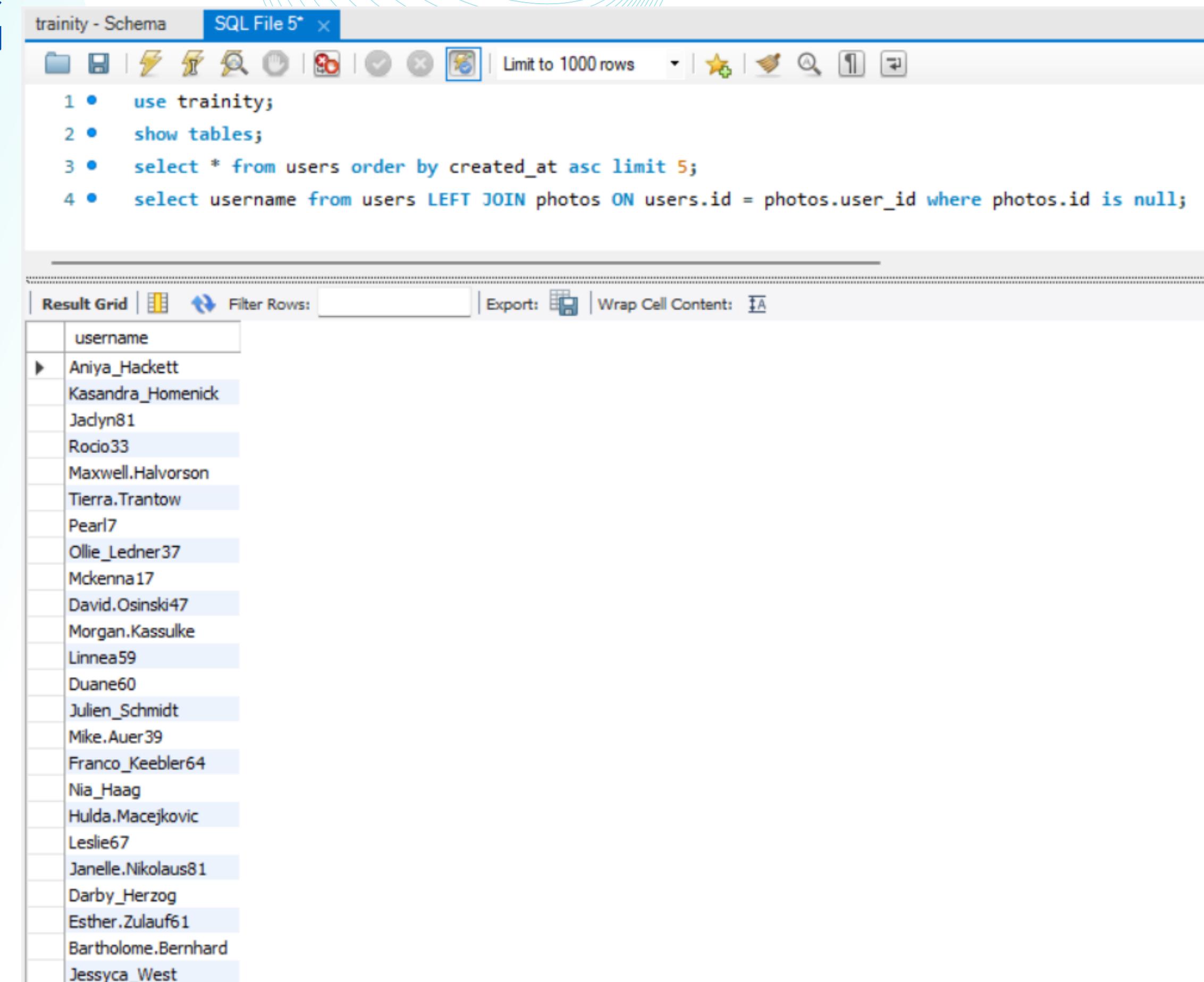
The screenshot shows a MySQL Workbench interface with the following details:

- SQL Editor:** The tab is labeled "trainity - schema". The toolbar includes icons for file operations, a lightning bolt (refresh), a magnifying glass (search), and a hand (refresh). A dropdown menu says "Limit to 1000 rows".
- Query History:** Four numbered steps are listed:
 - 1 • use trainity;
 - 2 • show tables;
 - 3
 - 4 • select * from users order by created_at asc limit 5;
- Result Grid:** The results of the final query are displayed in a grid. The columns are "id", "username", and "created_at". The data is as follows:

| | 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 |
| | HULL | HULL | HULL |

2. REMIND INACTIVE USERS TO START POSTING

To increase the user interaction marketing team reminds by sending mail or sending message to the users who have never posted in recent times.



The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • use trainity;
2 • show tables;
3 • select * from users order by created_at asc limit 5;
4 • select username from users LEFT JOIN photos ON users.id = photos.user_id where photos.id is null;
```

The result grid displays the 'username' column with the following data:

| username |
|---------------------|
| Aniya_Hackett |
| Kasandra_Homenick |
| Jadyn81 |
| Rocio33 |
| Maxwell.Halvorson |
| Tierra.Trantow |
| Pearl7 |
| Ollie_Ledner37 |
| Mckenna17 |
| David.Osinski47 |
| Morgan.Kassulke |
| Linnea59 |
| Duane60 |
| Julien_Schmidt |
| Mike.Auer39 |
| Franco_Keebler64 |
| Nia_Haag |
| Hulda.Macejkovic |
| Leslie67 |
| Janelle.Nikolaus81 |
| Darby_Herzog |
| Esther.Zulauf61 |
| Bartholome.Bernhard |
| Jessyca_West |

3. DECLARING CONTEST WINNER

To increase the user interaction marketing team starts contest in which the user with maximum likes on a single photo wins.

The screenshot shows a MySQL Workbench interface. The top tab bar has 'trainity - Schema' and 'SQL File 5*' selected. The toolbar includes icons for file operations, schema browser, and various database functions. The SQL editor contains the following code:

```
2 • show tables;
3 • select * from users order by created_at asc limit 5;
4 • select username from users LEFT JOIN photos ON users.id = photos.user_id where photos.id is null;
5 • select username, count(*) AS total from photos JOIN likes ON likes.photo_id = photos.id JOIN users ON photos.user_id = users.id
6 group by photos.id order by total desc LIMIT 1;
```

The fifth line of code is highlighted with a blue selection bar. Below the editor is a results grid with the following data:

| | username | total |
|---|---------------|-------|
| ▶ | Zack_Kemmer93 | 48 |

4. HASHTAG RESEARCHING

Hashtag usage helps to find the current trending topics and the brands use this data to find the current top trends so that brands can also use them.

trainity - Schema SQL File 5* ×

Limit to 1000 rows

```
3 ● select * from users order by created_at asc limit 5;
4 ● select username from users LEFT JOIN photos ON users.id = photos.user_id where photos.id is null;
5 ● select username, count(*) AS total from photos JOIN likes ON likes.photo_id = photos.id JOIN users ON photos.user_id = users.id
   group by photos.id order by total desc LIMIT 1;
7 ● select tag_name, count(*) as total from photo_tags join tags on photo_tags.tag_id = tags.id group by tags.id order by total desc limit 5;
```

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

| | tag_name | total |
|---|----------|-------|
| ▶ | smile | 59 |
| | beach | 42 |
| | party | 39 |
| | fun | 38 |
| | concert | 24 |

5. LAUNCH AD CAMPAIGN

To increase the value of the company ads are launched and to gain more profit teams look for the best day to launch ads (most active days). Here thursday and sunday

trainity - Schema SQL File 5* x

Limit to 1000 rows

```
4 ● select username from users LEFT JOIN photos ON users.id = photos.user_id where photos.id is null;
5 ● select username, count(*) AS total from photos JOIN likes ON likes.photo_id = photos.id JOIN users ON
6 group by photos.id order by total desc LIMIT 1;
7 ● select tag_name, count(*) as total from photo_tags join tags on photo_tags.tag_id = tags.id group by
8 ● tag_name order by total desc;
```

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

| | day | total |
|---|-----------|-------|
| ▶ | Thursday | 16 |
| | Sunday | 16 |
| | Friday | 15 |
| | Tuesday | 14 |
| | Monday | 14 |
| | Wednesday | 13 |
| | Saturday | 12 |

6. USER ENGAGEMENT

This is used to analyze the usage of id by calculating the average (total photos/total users). Investors use this data to check the value of the company.

trainity - Schema SQL File 5* ×

Limit to 1000 rows

```
5 • select username, count(*) AS total from photos JOIN likes ON likes.photo_id = p
6   group by photos.id order by total desc LIMIT 1;
7 • select tag_name, count(*) as total from photo_tags join tags on photo_tags.tag_
8 • select dayname(created_at) as day, count(*) as total from users group by day or
9 • select (select count(*) from photos)/(select count(*) from users) as average;
```

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

| | average |
|---|---------|
| ▶ | 2.5700 |

7. BOTS & FAKE ACCOUNTS

Investors want to know whether there are bots or fake accounts. If liking all the posts on the platform are bots which decrease the actual user engagement

trainity - Schema SQL File 5* ×

7 • select tag_name, count(*) as total from photo_tags join tags on photo_tags.tag_id = tags.id
8 • select dayname(created_at) as day, count(*) as total from users group by day order by total d
9 • select (select count(*) from photos)/(select count(*) from users) as average;
10 • select username, count(*) as total_likes from users join likes on users.id = likes.user_id
11 group by likes.user_id having total_likes = (select count(*) from photos);

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

| username | total_likes |
|--------------------|-------------|
| Aniya_Hackett | 257 |
| Jadyn81 | 257 |
| Rodo33 | 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 |

RESULT

While working on this project I used SQL WORKBENCH which helped me to understand new topics like joins, limit etc.

The main use of SQL was to retrieve the data according to the requirement.

This project helped me how to analyze the data and use WORKBENCH.

THANK YOU