PROJECT REPORT

Project Description

The project requires understanding of the data and answering business questions through SQL queries. Instagram user analytics is studied and useful insights are taken out to help in marketing and investments for the app. The database includes users, posted photos, photo tags, tags, likes, comments and follows. And from these, we have to answer certain questions that could improve the app business.

First, data understanding is important, followed by understanding the questions asked. Once understood, we can proceed to develop queries to solve the questions using SQL.

Approach

Steps taken to answer the questions to get the insights:

1. Understanding data thoroughly. This includes understanding each table separately and the contents inside.
2. Understanding the questions asked. This includes deeply diving into the data and see where the answer lies, which tables to be used to answer the specific question, etc.
3. Constructing the sql queries to get the right information we require and then validating them to ensure the results are correct.

Tech-Stack Used

MySQL Workbench 8.0 CE is the software used for the project. MySQL is easy and comfortable for these works which is the reason for its use in the project.

Insights

The insights found from the project are:

1. Oldest users all registered in May 2016.
2. Most users like to use positive and happiness related tags to their posts.
3. Thursday is the best day when registrations flock in the app as compared to other days.
4. Most users do not posts a lot and uses the app mostly for the entertainment.
5. There are a lot of bot accounts.

SQL Tasks:

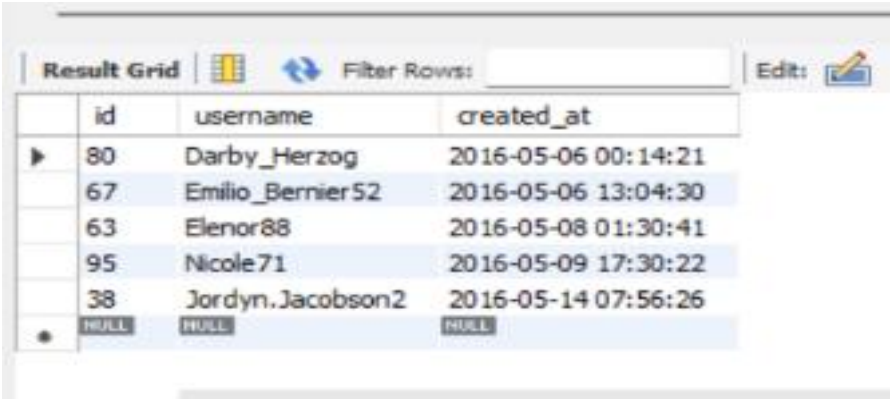
A) Marketing Analysis:

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

QUERY:

```
SELECT
        id, username, created_at
FROM
   ig_clone.users ORDER BY
   created_at  ASC
LIMIT 5;
```

OUTPUT:

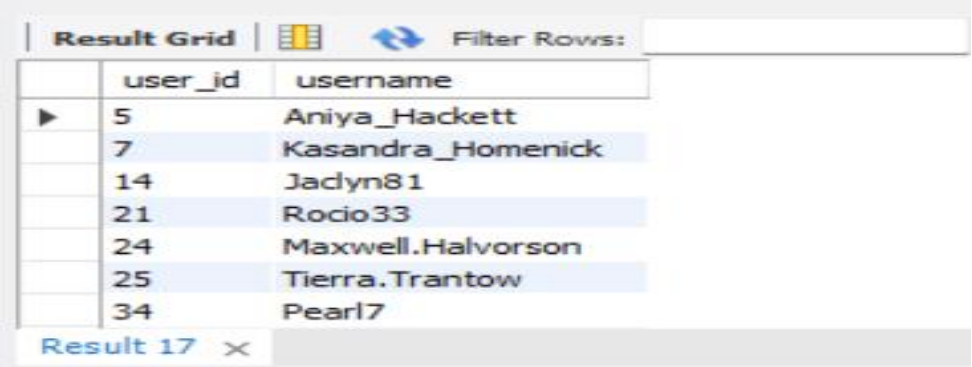


2. Identify users who have never posted a single photo on Instagram.

QUERY:

```
SELECT
   u.id AS user_id,
   u.username
FROM
   users u
LEFT JOIN
   photos p
ON
   u.id = p.user_id
WHERE
   p.user_id IS NULL;
```

OUTPUT:



Result Grid | Filter Rows:

user_id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jadyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7

Result 17

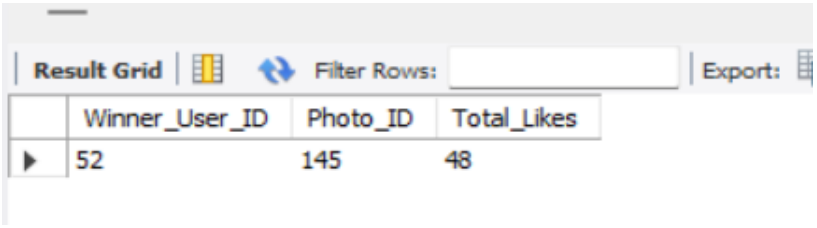
3. Determine the winner of the contest and provide their details to the team.

QUERY:

```
SELECT
   p.user_id AS Winner_User_ID,
   l.photo_id AS Photo_ID,
   COUNT(l.photo_id) AS Total_Likes
FROM
```

```
    photos p
JOIN
    likes l
ON
    p.id = l.photo_id
GROUP BY
    l.photo_id, p.user_id
ORDER BY
    Total_Likes DESC
LIMIT 1;
```

OUTPUT:



	Winner_User_ID	Photo_ID	Total_Likes
▶	52	145	48

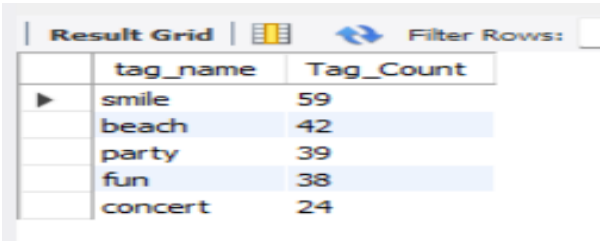
4. Identify and suggest the top five most commonly used hashtags on the platform.

QUERY:

```
SELECT
    t.tag_name,
    COUNT(pt.tag_id) AS Tag_Count
FROM
    photo_tags pt
JOIN
    tags t
ON
    pt.tag_id = t.id
GROUP BY
    t.tag_name
```

```
ORDER BY
    Tag_Count DESC
LIMIT 5;
```

OUTPUT:



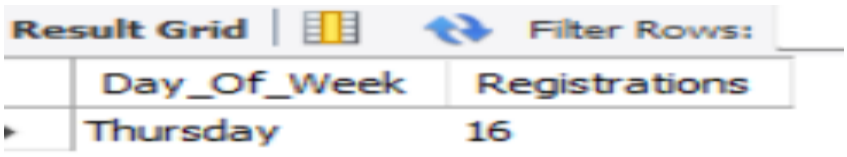
tag_name	Tag_Count
smile	59
beach	42
party	39
fun	38
concert	24

5. Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

QUERY:

```
SELECT
    DAYNAME(created_at) AS Day_Of_Week,
    COUNT(*) AS Registrations
FROM
    users
GROUP BY
    DAYNAME(created_at)
ORDER BY
    Registrations DESC
LIMIT 1;
```

OUTPUT:



Day_Of_Week	Registrations
Thursday	16

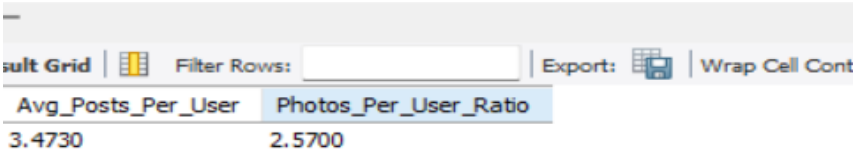
B) Investor Metrics:

1. 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.

QUERY:

```
WITH Photo_Counts AS (
    SELECT
        user_id,
        COUNT(*) AS Photo_Count
    FROM
        photos
    GROUP BY
        user_id
)
SELECT
    AVG(Photo_Count) AS Avg_Posts_Per_User,
    (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users)
AS Photos_Per_User_Ratio
FROM
    Photo_Counts;
```

OUTPUT:



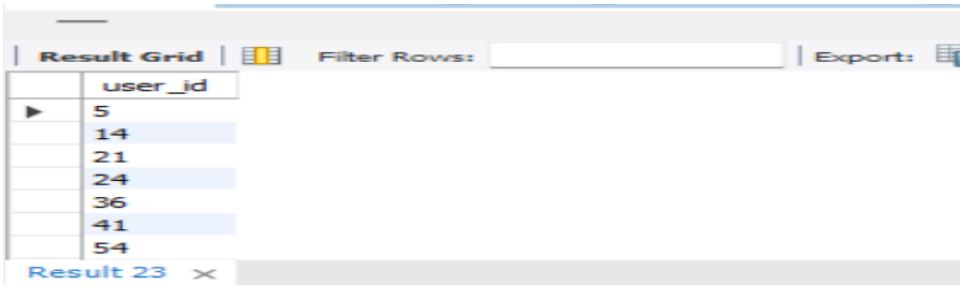
2. Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

QUERY:

```
WITH Total_Photos AS (
```

```
    SELECT COUNT(*) AS Total_Photos_Count
    FROM photos
),
User_Photo_Likes AS (
    SELECT
        user_id,
        COUNT(DISTINCT photo_id) AS Liked_Photos_Count
    FROM
        likes
    GROUP BY
        user_id
)
SELECT
    u.user_id
FROM
    User_Photo_Likes u
JOIN
    Total_Photos t
ON
    u.Liked_Photos_Count = t.Total_Photos_Count;
```

OUTPUT:



Result

From this project, I have learnt how to analyse data and use logic to develop codes to answered the desired questions. The results can be used to derive useful insights that can further be used for other purposes to improve the usage of the app.