Case Study: 2(Instagram User Analytics)

Description: To track how users engage and interact with our digital product by using user analysis process. The results can be used to derive business insights for marketing, product and development teams. Such insights will be used by various teams across the business

- to launch a new marketing campaign
- decide on features to build for an app
- track success of the app by measuring user engagement
- improve the experience altogether while helping the business grow

Plan: The project has been divided into two sections as there are two separate needs

- **Marketing:** As the marketing team wants to launch some campaigns, hence they require the data for following purposes
  - Rewarding the people who have been using the app for the longest time.
  - Reminding inactive users to start posting by sending them promotional mails to post their first photo.
  - Declaring the contest winner of the most liked single photo contest and check who got the most likes.
  - How to help a partner brand reach the most people on the platform using hashtags?
  - Launching an AD campaign on the best day possible.

- **Investor Metrics:** The investors wanted to know if the app is performing well unlike Facebook, which has become redundant nowadays. They wanted to assess the app on the following grounds
  - How many users are still active and posting on Instagram?
  - Is the app is getting crowded with fake and dummy accounts?

Prepare/Approach: Now we need to figure out our objectives on how to successfully fulfil the requirements. I'll be using MySQL workbench 8.0 CE as the UI is very clean and pretty easy to navigate. Also, it has separate sections to write queries, check errors and collect output in a single window.

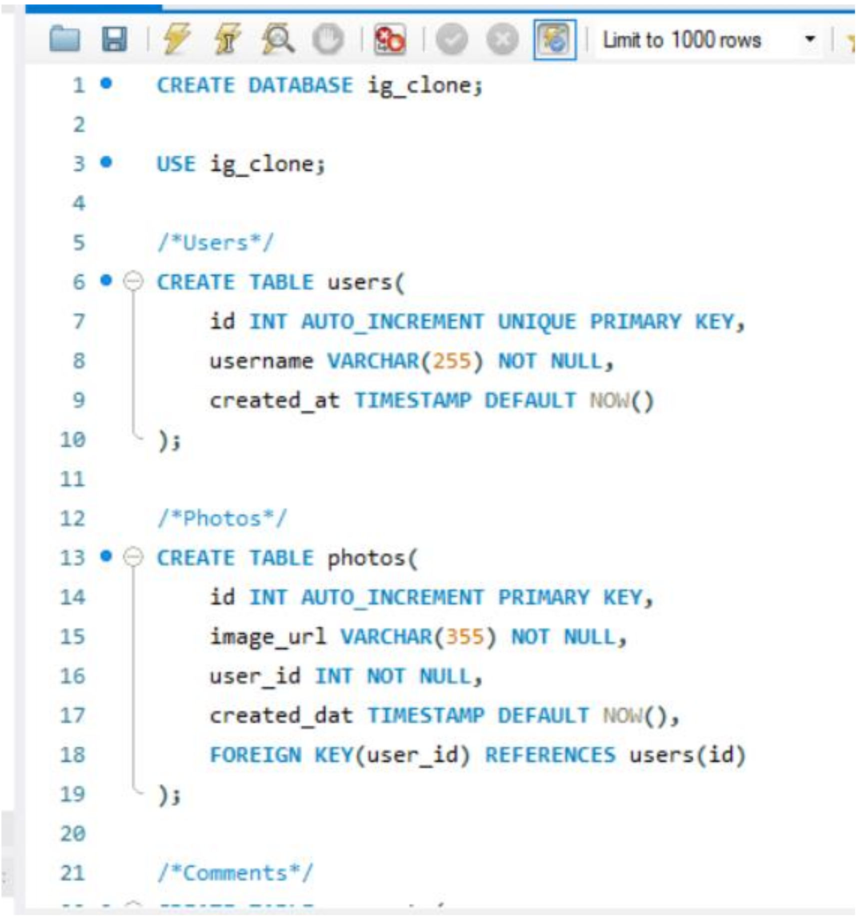
Hereby, I'll be creating a roadmap on how to fulfil the expectations of the marketing team and investors separately.

- **For marketing team:**
  - Rewarding Most Loyal users: We need to find the oldest users of the Instagram.
  - Reminding Inactive Users: We have to find the users who have never posted a single photo.

  - Declaring Contest Winner: We need to identify the winner of the contest based on the number of likes per photo and provide their details to the team.
  - Hashtag Researching: We need to identify and suggest the top 5 most commonly used hashtags on the platform.
  - Launch AD Campaign: We need to find the perfect day of the week when most users register on.
- **Investor Metrics:**
  - User Engagement: We need to provide how many times does average user post on Instagram along with the total number of photos on Instagram per total number of users.
  - Bots & Fake Accounts: We need to provide data on bots who have liked every single photo on the site.

Process: Now that we know the objectives and target, for continuing our process we’ll be requiring to create a database using which we can carry out various SQL queries.

- **Creating the database:** Since the database has already been provided under the name “Dataset”.
- **Opening the dataset:** Lets open the dataset and copy all the contents
- **Loading the dataset:** Lets open MySQL Workbench and paste the contents and click EXECUTE.



```
CREATE DATABASE ig_clone;

USE ig_clone;

/*Users*/
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
    user_id INT NOT NULL,
    created_dat TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);

/*Comments*/
```

Analyze and Share: As the database and tables has been created. Now finally I can start analyzing the tasks given, step by step.

At first, I'll try to complete the tasks related to marketing department.

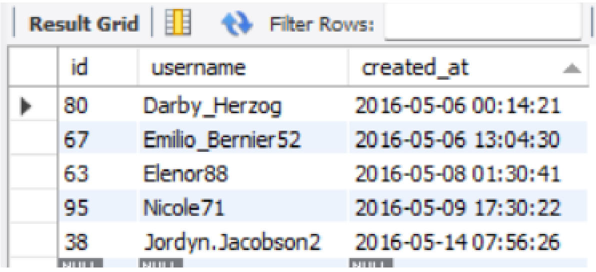
- **Rewarding Most Loyal Users:** To find the oldest users of the app, I need to sort the created_at column in ascending order. Hence, the following SQL query is used

*SELECT * FROM users*

ORDER BY created_at

LIMIT 5;

Upon execution, we'll get the following names along with the date and time in ascending order



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

- **Reminding Inactive Users:** To find the inactive users I need to find those people who never uploaded a single photo. Hence, under the influence of the id column from users table and the user_id column of the photos table and later by checking if the photo id is null or not, I can determine the inactive users. The following query is used

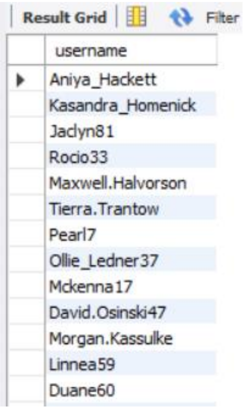
SELECT username

FROM users

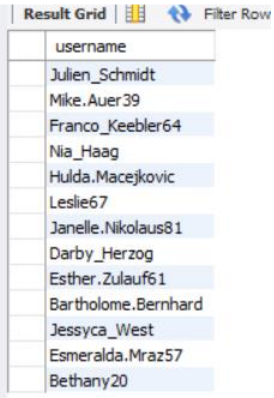
LEFT JOIN photos ON users.id = photos.user_id

WHERE photos.id IS NULL;

Upon execution we get the following results



username
Aniya_Hackett
Kasandra_Homenick
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60

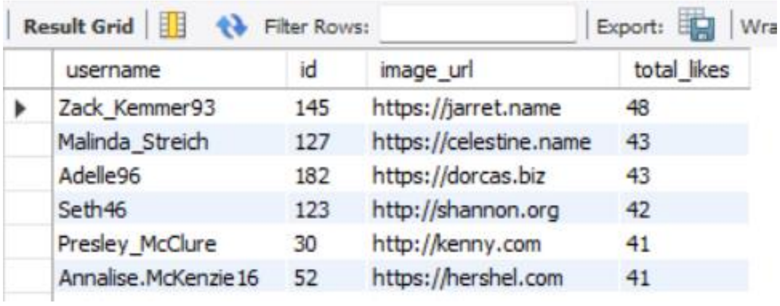


username
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

- **Declaring Contest Winner:** To get the winner I'll be requiring the data from three tables i.e, users, photos and likes. To count the number of likes, I'll be using COUNT() function and later arrange them in descending order. The following query has been written

```
SELECT users.username, photos.id, photos.image_url, COUNT(*) AS total_likes
FROM photos
INNER JOIN likes ON likes.photo_id = photos.id
INNER JOIN users ON photos.user_id = users.id
GROUP BY photos.id
ORDER BY total_likes DESC
```

Upon execution, the following output is taken



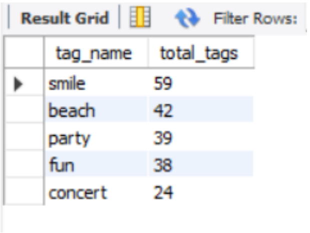
username	id	image_url	total_likes
Zack_Kemmer93	145	https://jarret.name	48
Malinda_Streich	127	https://celestine.name	43
Adelle96	182	https://dorcas.biz	43
Seth46	123	http://shannon.org	42
Presley_McClure	30	http://kenny.com	41
Annalise.McKenzie16	52	https://hershel.com	41

Hence, Zack_Kemmer93 is the clear winner with 48 total likes

- **Hashtag Researching:** Here we have to check the names of the top 5 hashtags that has been used by the users. For that we have to consider two tables i.e, tags and photo_tags and later joining them using the id column of tags table with the tag_id column of photo_tags table. Hence the following query can be used to check the total tags

```
SELECT tag_name, COUNT(tag_name) AS total_tags
FROM tags
JOIN photo_tags ON tags.id = photo_tags.tag_id
GROUP BY tags.id
ORDER BY total_tags DESC
LIMIT 5;
```

Hence by counting the number of tag names and grouping it by tag id, we finally got the following output



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

- **Launch AD Camgaign:** The team wants to know the specific day to launch Ads. Hence, here I'll have to check on which day of the week people register on mostly. For that I'll be using the dayname() function to determine the day of the week from created_at column and then check for total registration using count() function. Hence, the following query is generated

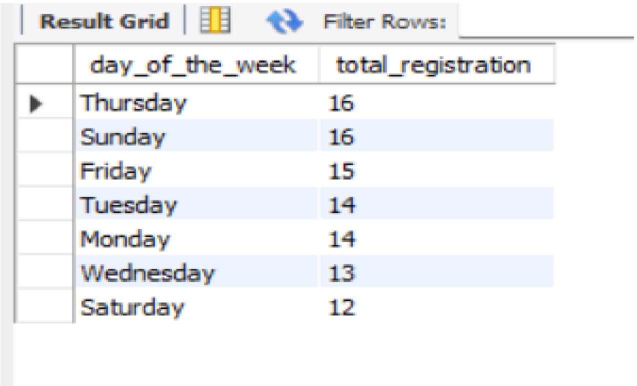
SELECT DAYNAME(created_at) AS day_of_the_week, COUNT(*) AS total_registration

FROM users

GROUP BY day_of_the_week

ORDER BY total_registration DESC;

From the following output we can see that Thursday and Sunday are two of the days when people mostly register. Therefore, the team can launch their ad campaigns on either Thursday or Sunday.



day_of_the_week	total_registration
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

By here I've completed my tasks for the marketing team. Now I'll be moving forward with the investor's requirements.

For the Investor Metrics,

- **User Engagement:** Basically, here they want to know whether the users are active and posting on Instagram or not. Since there are more than one requirements to be fulfilled, therefore I'll be dividing it into multiple sections for the ease of understanding.
  - **Average posts per user:** For this I'll be dividing total number of photos posted with total number of users. I'll be using the count function to determine the total number for both the cases. The following query is used

```
SELECT ((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users)) as
avg_user_post
```

Hence, the average user post is 2.57

- **Total users on Instagram:** Here we have two section of people who have never posted and the one who atleast posted once.
  At first, I'll find the number of people who have at least posted once. I'll be using Distinct statement to filter out the duplicates.

```
SELECT COUNT(DISTINCT(users.id)) AS total_number_of_users_with_posts
FROM users
JOIN photos ON users.id = photos.user_id;
```

The total number of users with atleast one post turned out to be 74.

Now, I'll find the total number of users who have never posted a single post.

```
SELECT COUNT(username) as total_users_with_no_post
FROM users
LEFT JOIN photos ON users.id = photos.user_id
WHERE photos.id IS NULL;
```

The total number of users with not even a single post is 26.

Therefore, adding them we can get the total number of users i.e, 100.

- **Total number of photos on Instagram:** Here I'll be using the COUNT statement to determine the number of posts from the photos table

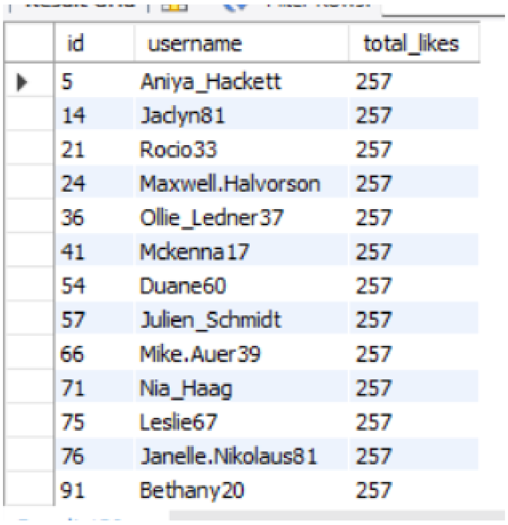
```
SELECT COUNT(photos.image_url) as total_posts
from photos
```

Hence, the total number of photos on Instagram is 257.

- **Bots and Fake Accounts:** Basically, they wanted to know whether the app is filled with bots and fake account. To determine this I'll be checking the number of users who have liked almost all the posts since it's not possible for a normal user to do so. Since WHERE cannot be used for aggregate functions therefore I'll be using HAVING clause. Also, we already know that the total number of photos on Instagram is 257. Therefore, the following query is used

```
SELECT users.id,username, COUNT(users.id) As total_likes
FROM users
JOIN likes ON users.id = likes.user_id
GROUP BY users.id
HAVING total_likes = 257;
```

The following output table consists of the names which might probably be the bot accounts as they have liked all 257 photos.



id	username	total_likes
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

Insights and Results: This project provided enough knowledge to me about the industry needs and assessments.

Previously, I had no idea how data related industry works. I'll be using some important points to provide an insight on my project completion.

1. **Learning experience:** This project has been significant milestone for me as it has provided a distinct step by step process of doing a project oderly and completing it under or before the time given. There have been many challenges that I encountered for eg: While doing the investor metrics for bot accounts, I got stuck because I was using WHERE clause instead of HAVING clause. Doing some digging online I came to know that WHERE clause cannot be used with aggregated functions. Also I had a big confusion using the JOIN commands which got cleared while watching the learning videos and doing this project together.
   This project cleared the almost every basic concepts that was very essential to begin this journey of being an Analyst.
2. **Time Management:** The time given for this project was enough for me as I already had a bit of basic knowledge of SQL. Even though I wanted to complete it as soon as possible so that I could practice with other SQL questions available. This projects gave me a glimpse of managing time and completing the project within the give timeframe. The most critical skill in this industry is without any doubt is Time

management. By doing this project and submitting it before time frame I was able to achieve my goal of time management.

3. **Attention to detail:** The way SQL is designed, requires a high level of attention to detail. This language attracts highest level of concentration for a beginner like me. One single mistake and whole output will be compromised. For a beginner like me, it is very essential to keep an eye on the names and words regularly. Completing this project on time strengthens my capability of attention to details.
4. **Collaborations:** It would be really unfair if I don't mention the websites who helped me completing this project by providing the essential concepts. I find these websites to be very helpful for anyone with basic language understanding.
    a. W3Scools: https://www.w3schools.com/sql/
    b. SQL Tutorial: https://www.sqltutorial.org/
    c. Trainity Learning: https://trainity.link/data/learning
5. **Practical application:** Completing this project gave me an idea of the process of applying the theoretical knowledge from the videos and pdfs to the real-world scenarios. At the end it gave me a deeper understanding of the tools and language that has been used for the completion of this project.

Overall, completing this project have enhanced the process of my learning experience. It was really interesting and challenging at the same time. By completing this project on time, I gained a bit of confidence of moving further towards my journey of being an Analyst.

Tech-Stack used: The following softwares has been used during the process

1. My SQL Workbench 8.0 CE
2. Microsoft Word 2019

Drive Link:

https://drive.google.com/drive/folders/16o-EQhXx3339ZOF6zLno4vRRmOz4vCWM

under the name "project_2"

Prepared by: Sushant Karmakar