PROJECT 2: Instagram User Analytics

Project Description:

User analysis is the process by which we track how users engage and interact with our digital product (software or mobile application) in an attempt to derive business insights for marketing, product & development teams. These insights are then used by teams across the business to launch a new marketing campaign, decide on features to build for an app, track the success of the app by measuring user engagement and improve the experience altogether while helping the business grow.

In this project we are working with the product team of Instagram and the product manager has asked some insights from the dataset by asking some questions.

There are 5 questions from the marketing team that need analysis and answering (my task) and 2 questions from the investors.

Approach:

For completing this project, first I finished the course material provided to us. Learnt about SQL and how to write a simple query. Further, I learnt about various operations and functions that can be performed on a dataset to gain insights.

I started out by reading the description of the project and then loaded the dataset into db-fiddle. After that I solved each question individually by running queries over the data and storing the results. I had to do some search and read about some new functions for answering some of the problems.

Tech stack used:

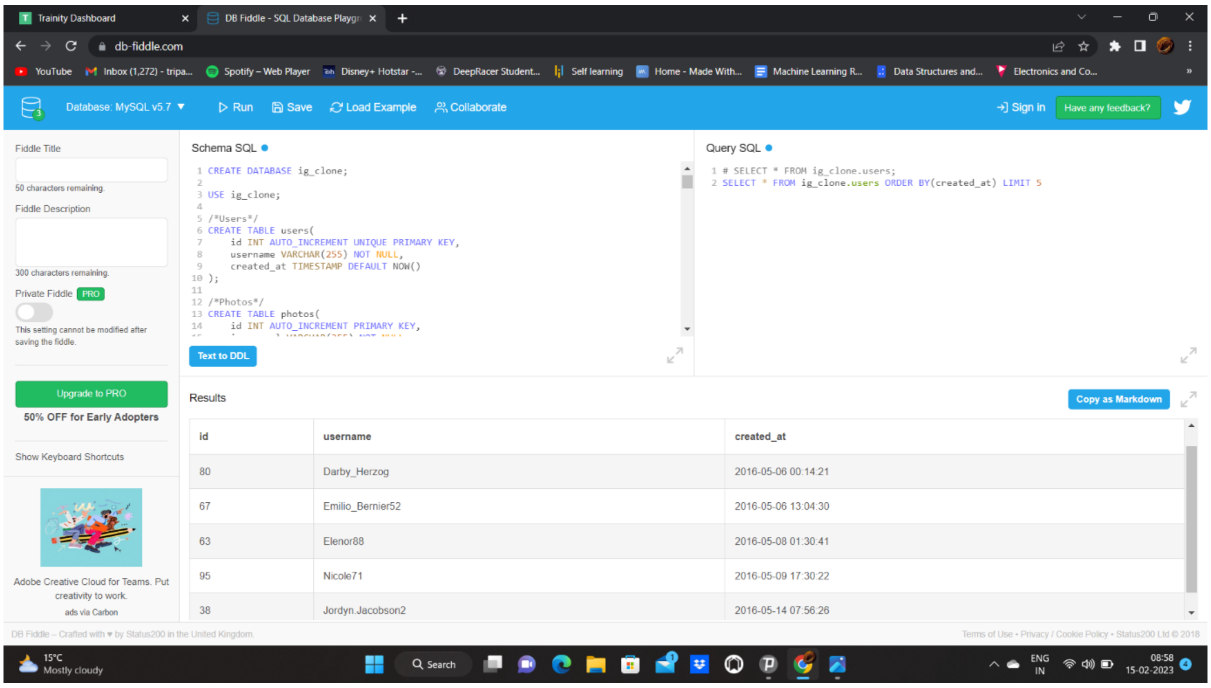
Online tool: db-fiddle

Link: Fiddle Link

Insights:

1.1Rewarding Most Loyal Users: People who have been using the platform for the longest time

Query: SELECT * FROM ig_clone.users ORDER BY(created_at) LIMIT 5



Schema SQL

```
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,
```

Query SQL

```
# SELECT * FROM ig_clone.users;
SELECT * FROM ig_clone.users ORDER BY(created_at) LIMIT 5
```

Results

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

People with oldest records belong to the year 2016 and all 5 of them created their account in the month of May.

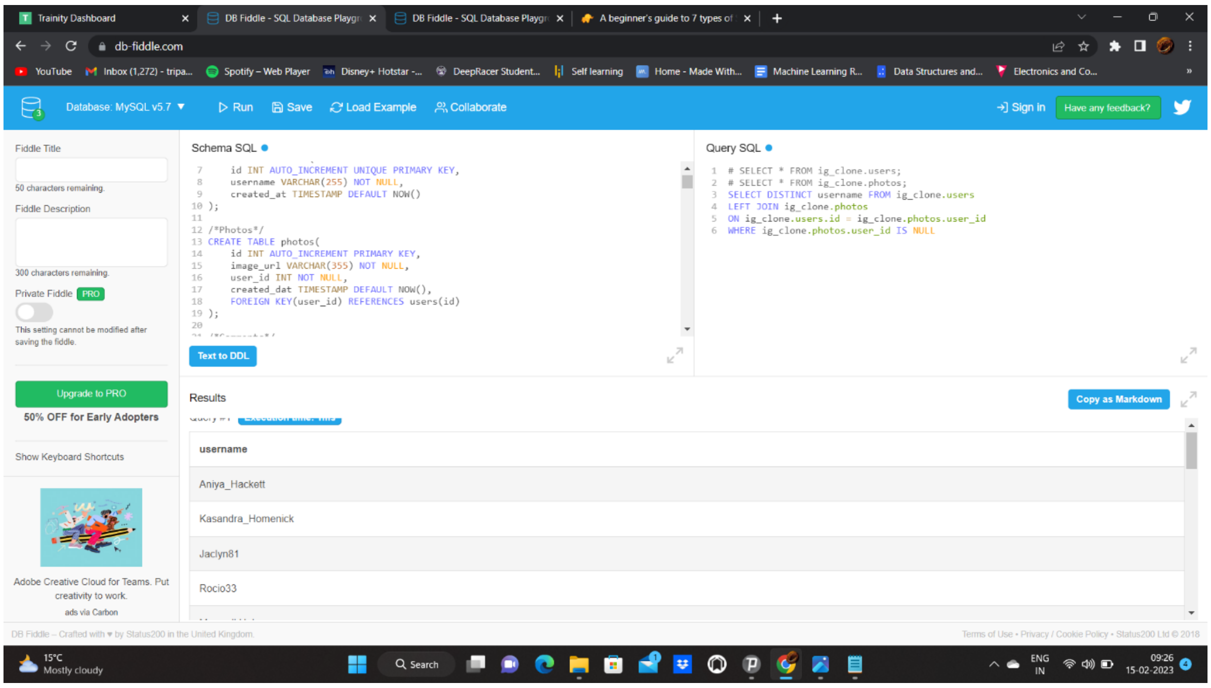
1.2 Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.
Your Task: Find the users who have never posted a single photo on Instagram

Query: SELECT DISTINCT username FROM ig_clone.users

LEFT JOIN ig_clone.photos

ON ig_clone.users.id = ig_clone.photos.user_id

WHERE ig_clone.photos.user_id IS NULL


Schema SQL

```
    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)
);
```

Query SQL

```
# SELECT * FROM ig_clone.users;
# SELECT * FROM ig_clone.photos;
SELECT DISTINCT username FROM ig_clone.users
LEFT JOIN ig_clone.photos
ON ig_clone.users.id = ig_clone.photos.user_id
WHERE ig_clone.photos.user_id IS NULL
```

Results

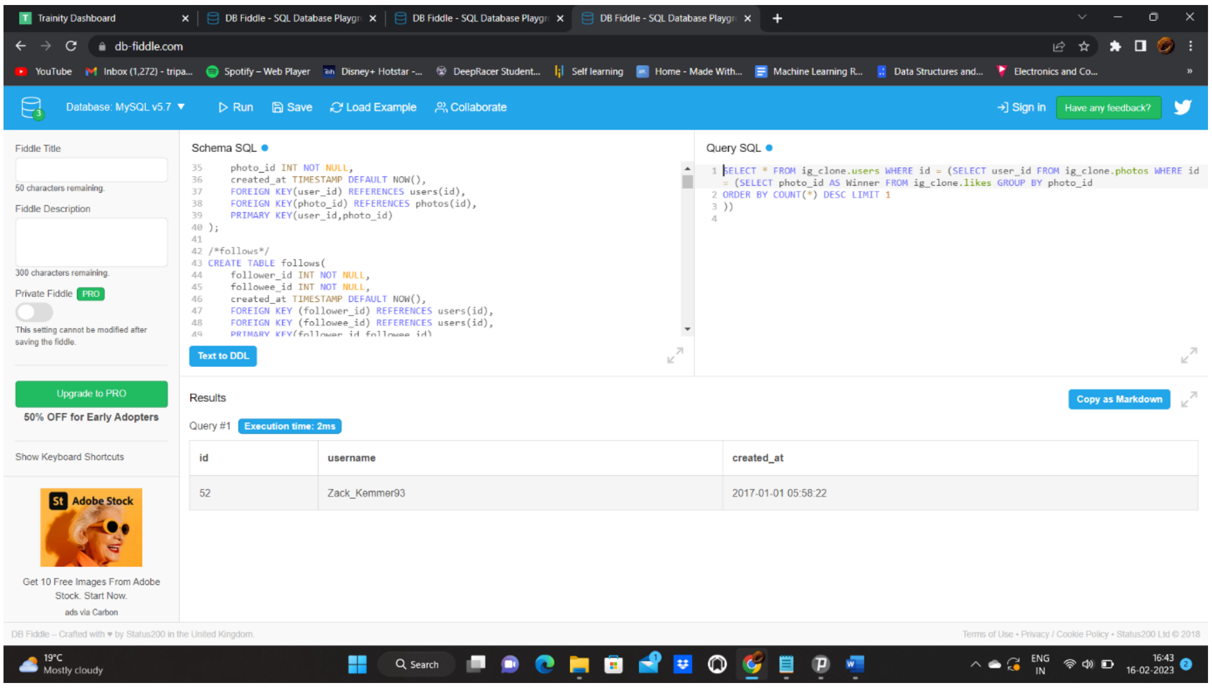
username
Aniya_Hackett
Kasandra_Homenick
Jaclyn81
Rocio33

We now have all the users who are inactive from the time of the account creation i.e we have the username and now the marketing team can send them mails or ads regarding posting something on the IG.

2.3 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

Query: SELECT photo_id AS Winner FROM ig_clone.likes GROUP BY photo_id
ORDER BY COUNT(*) DESC LIMIT 1



Schema SQL

```
    photo_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id),
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    PRIMARY KEY(user_id,photo_id)
);

/*follows*/
CREATE TABLE follows(
    follower_id INT NOT NULL,
    followee_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY (follower_id) REFERENCES users(id),
    FOREIGN KEY (followee_id) REFERENCES users(id),
    PRIMARY KEY(follower_id,followee_id)
```

Query SQL

```
SELECT * FROM ig_clone.users WHERE id = (SELECT user_id FROM ig_clone.photos WHERE id
= (SELECT photo_id AS Winner FROM ig_clone.likes GROUP BY photo_id
ORDER BY COUNT(*) DESC LIMIT 1
))
```

Results

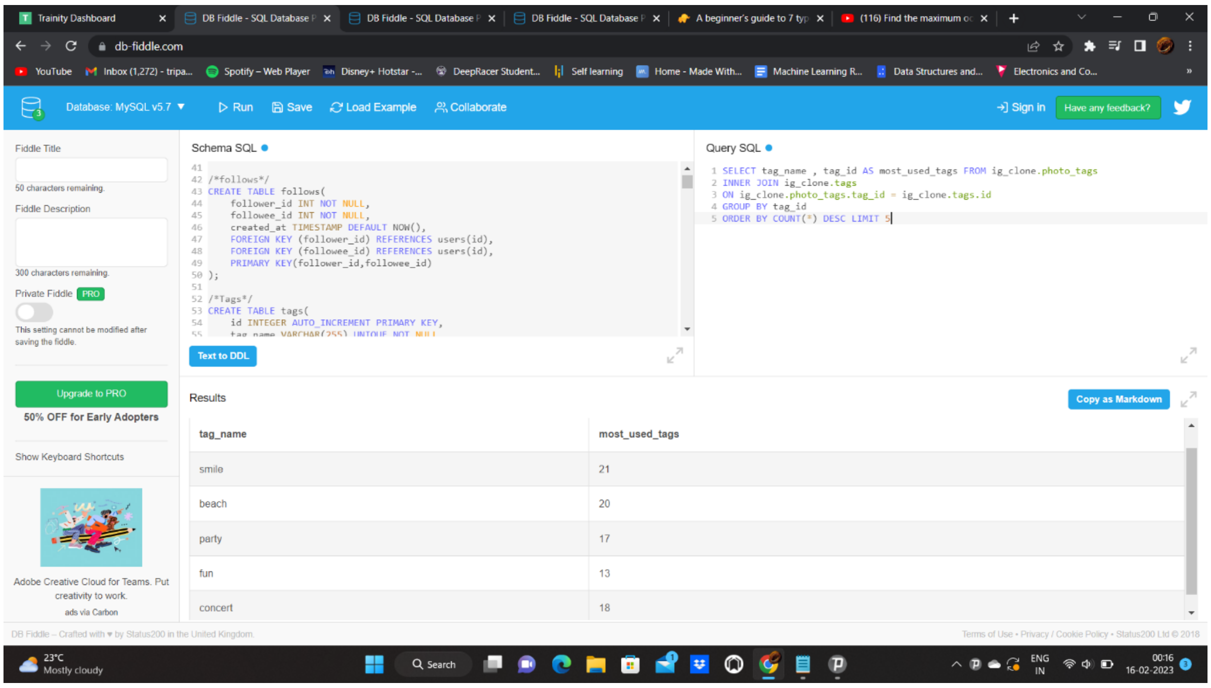
Query #1 Execution time: 2ms

id	username	created_at
52	Zack_Kemmer93	2017-01-01 05:58:22

2.4 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

Query: SELECT tag_name , tag_id AS most_used_tags FROM ig_clone.photo_tags
INNER JOIN ig_clone.tags
ON ig_clone.photo_tags.tag_id = ig_clone.tags.id
GROUP BY tag_id
ORDER BY COUNT(*) DESC LIMIT 5



Schema SQL

```
/*follows*/
CREATE TABLE follows(
    follower_id INT NOT NULL,
    followee_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY (follower_id) REFERENCES users(id),
    FOREIGN KEY (followee_id) REFERENCES users(id),
    PRIMARY KEY(follower_id,followee_id)
);

/*Tags*/
CREATE TABLE tags(
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
```

Query SQL

```
SELECT tag_name , tag_id AS most_used_tags FROM ig_clone.photo_tags
INNER JOIN ig_clone.tags
ON ig_clone.photo_tags.tag_id = ig_clone.tags.id
GROUP BY tag_id
ORDER BY COUNT(*) DESC LIMIT 5
```

Results

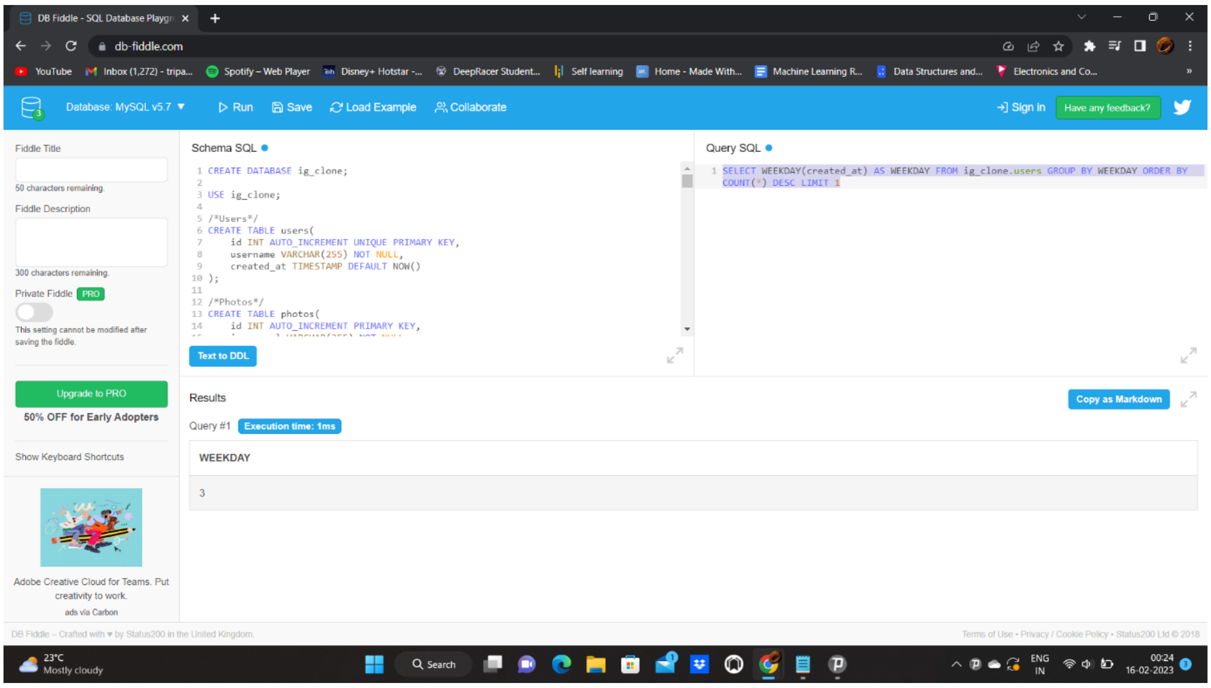
tag_name	most_used_tags
smile	21
beach	20
party	17
fun	13
concert	18

The team now has the most used tag names on the IG. People have posted more pictures having theme smile, beach, party, fun, concert. These tags suggest that people have posted more when going out for fun things like party or concert or beach.

2.5Launch 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

Query: SELECT WEEKDAY(created_at) AS WEEKDAY FROM ig_clone.users GROUP BY WEEKDAY ORDER BY COUNT(*) DESC LIMIT 1



Schema SQL

```sql
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,
```

Query SQL

```sql
SELECT WEEKDAY(created_at) AS WEEKDAY FROM ig_clone.users GROUP BY WEEKDAY ORDER BY
COUNT(*) DESC LIMIT 1
```

Results

Query #1 Execution time: 1ms

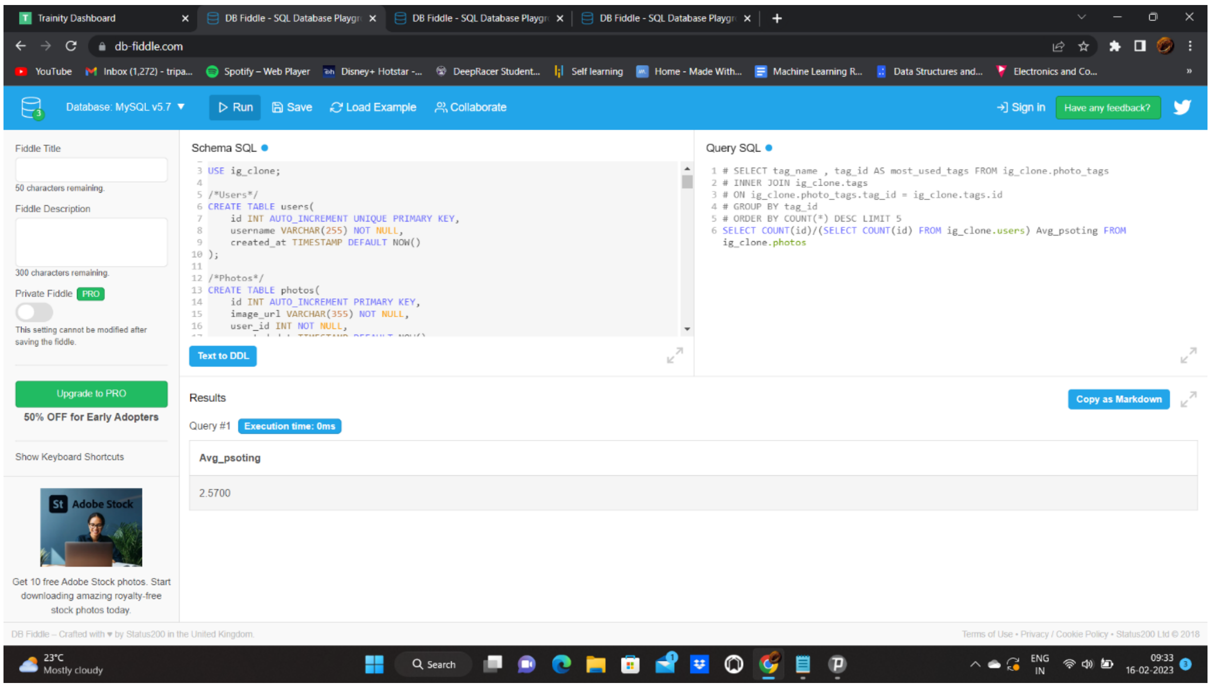
WEEKDAY
3

The result 3 here means the 4th day of the week i.e. Thursday is the day when most of the users registered on this means people are likely to spend more time on Thursday on IG than any other day so Thursday could be the best possible day to show ads regarding anything to attract the people towards it.

2.1**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

Query: SELECT COUNT(id)/(SELECT COUNT(id) FROM ig_clone.users) Avg_psoting FROM ig_clone.photos


Schema SQL

```
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,
```

Query SQL

```
# SELECT tag_name , tag_id AS most_used_tags FROM ig_clone.photo_tags
# INNER JOIN ig_clone.tags
# ON ig_clone.photo_tags.tag_id = ig_clone.tags.id
# GROUP BY tag_id
# ORDER BY COUNT(*) DESC LIMIT 5
SELECT COUNT(id)/(SELECT COUNT(id) FROM ig_clone.users) Avg_psoting FROM
ig_clone.photos
```

Results

Query #1 Execution time: 0ms

Avg_psoting
2.5700

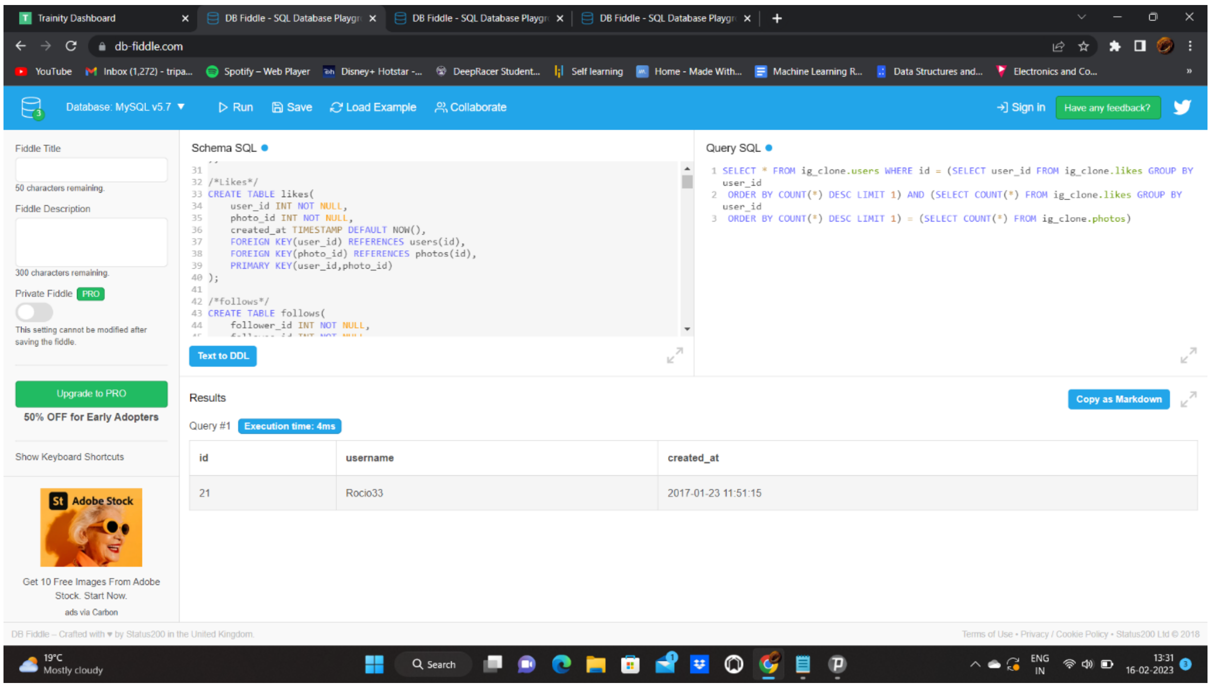
2.2**Bots & Fake Accounts:** The investors want to know if the platform is crowded with fake and dummy accounts

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).

Query: SELECT * FROM ig_clone.users WHERE id = (SELECT user_id FROM ig_clone.likes GROUP BY user_id

ORDER BY COUNT(*) DESC LIMIT 1) AND (SELECT COUNT(*) FROM ig_clone.likes GROUP BY user_id

ORDER BY COUNT(*) DESC LIMIT 1) = (SELECT COUNT(*) FROM ig_clone.photos)



Schema SQL

```sql
/*Likes*/
CREATE TABLE likes(
    user_id INT NOT NULL,
    photo_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id),
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    PRIMARY KEY(user_id,photo_id)
);

/*follows*/
CREATE TABLE follows(
    follower_id INT NOT NULL,
```

Query SQL

```sql
SELECT * FROM ig_clone.users WHERE id = (SELECT user_id FROM ig_clone.likes GROUP BY user_id
 ORDER BY COUNT(*) DESC LIMIT 1) AND (SELECT COUNT(*) FROM ig_clone.likes GROUP BY user_id
 ORDER BY COUNT(*) DESC LIMIT 1) = (SELECT COUNT(*) FROM ig_clone.photos)
```

Results

Query #1 Execution time: 4ms

id	username	created_at
21	Rocio33	2017-01-23 11:51:15

The account with the user_id = 21 is likely to be bot since this account has liked each and every photo present on Instagram this beyond the scope of any human.

<u>Result:</u>

While making the project I got to apply SQL that I learned form the platform into real world problems to gain insights. I got to work with the data for the first time and felt really good when results showed up after running the query.

This project helped me to strengthen my concepts of SQL and now I can confidently move on to the next topics knowing that this one is well done.