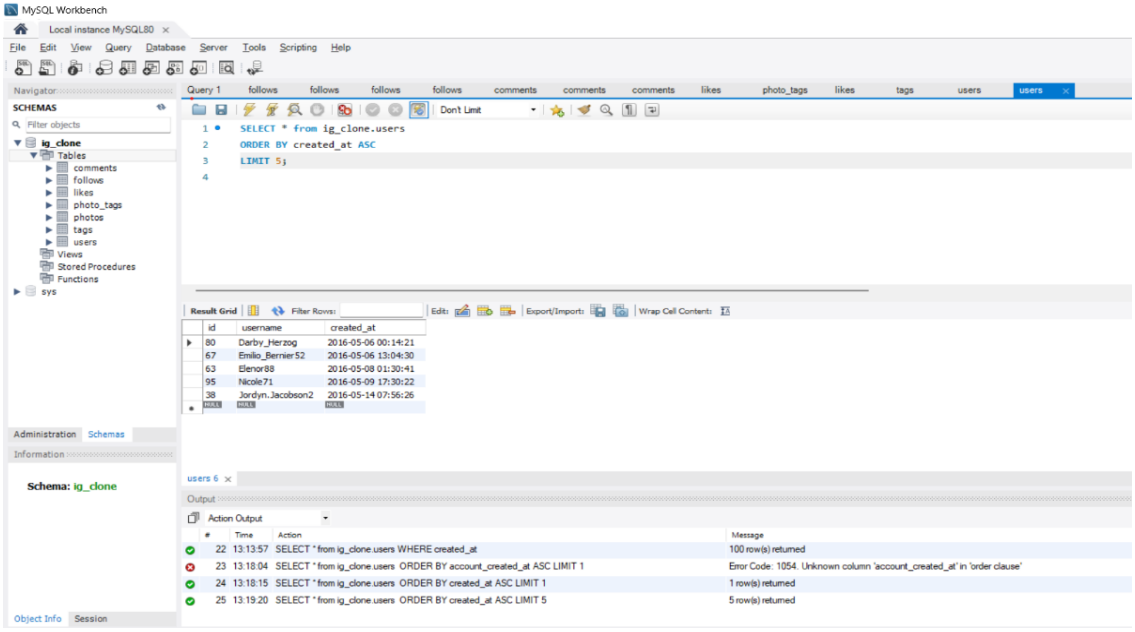
Instagram user Analytics Project-2

(Trainity Assignment)

SQL Task:

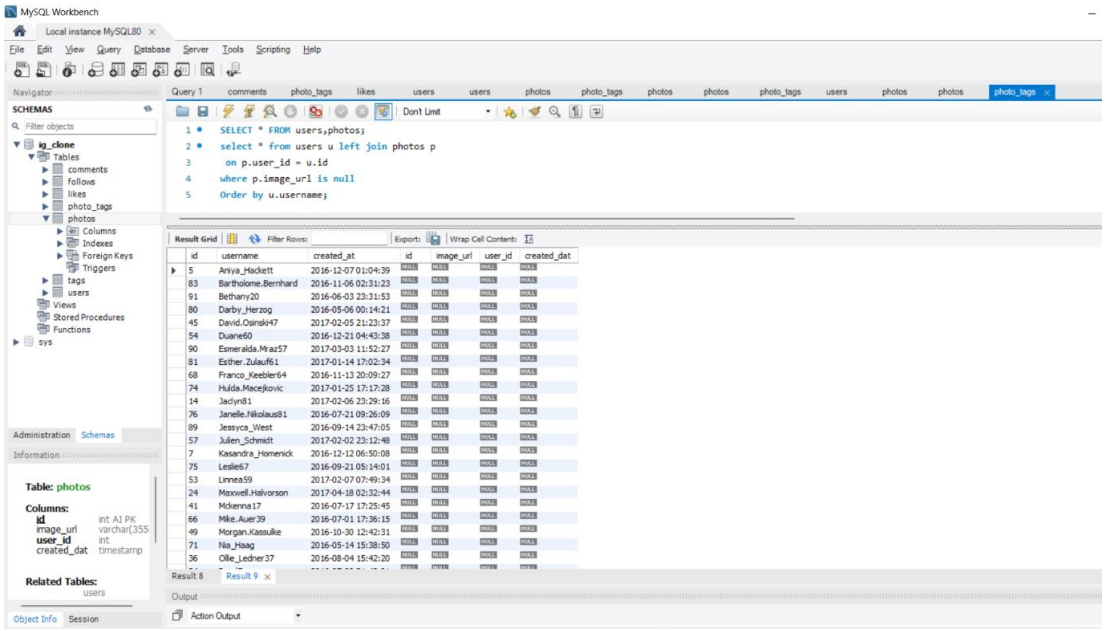
A) Marketing Analysis:

1. **Loyal user reward:** Identify the five oldest users on Instagram from the provided data.



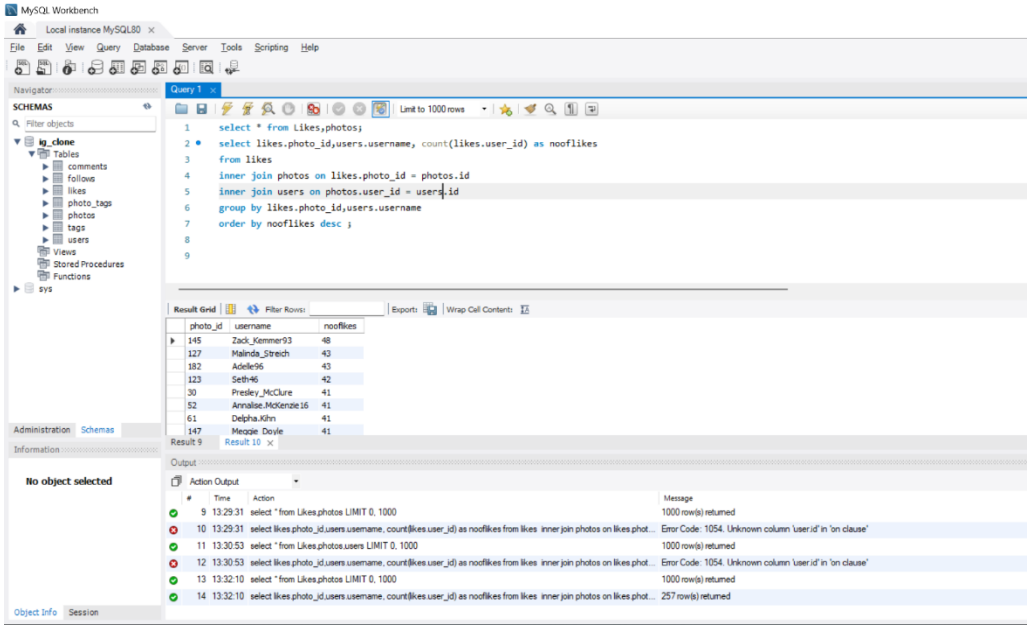
Answer: In this task we first use the ig_clone database ‘s table users to identify the five oldest users using select query. and arrange the data according to the created_at (date of account creation) in ascending order. And limit the data up to 5 because we need the 5 oldest users.

2. **Inactive User Engagement:** Identify users who have never posted a single photo on Instagram.



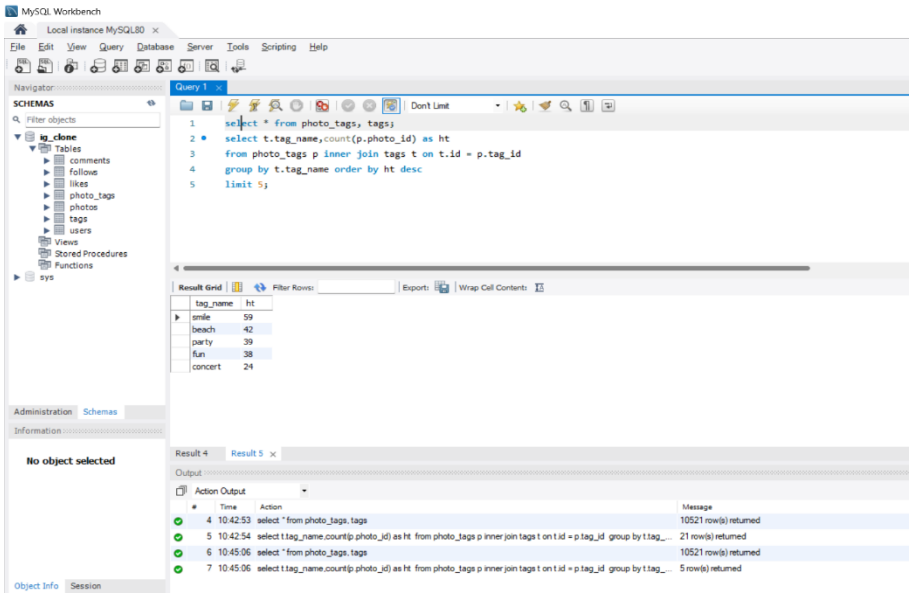
Answer: In this, task we will ned the users table and photos table from database Ig_clone.
Using the select query we join the user (u) and Photos (p) and photo's user_id will be equal to user's id.Using WHERE clause we will set Image_url as null (to find who have zero post) and arrange in the order by username so we will get users name who have posted zero post.

3. **Contest Winner Declaration:** Determine the winner of the contest and provide their details to the team. (find the most liked single photo).



Answer: In this, task we will need likes, users and photos table from database.
Using Select query we will fetch Photo_id from likes & username from user's table.
using Count clause we will count the user_id as no. of likes from likes table.
now, we will inner join likes, photos and users. where from Likes's photo_id is equal to photo's id and photo's user_id equal to user's id. Now we will group them by photo_id and username and arrange them in order by no of likes in descending. To get the most liked single photo.

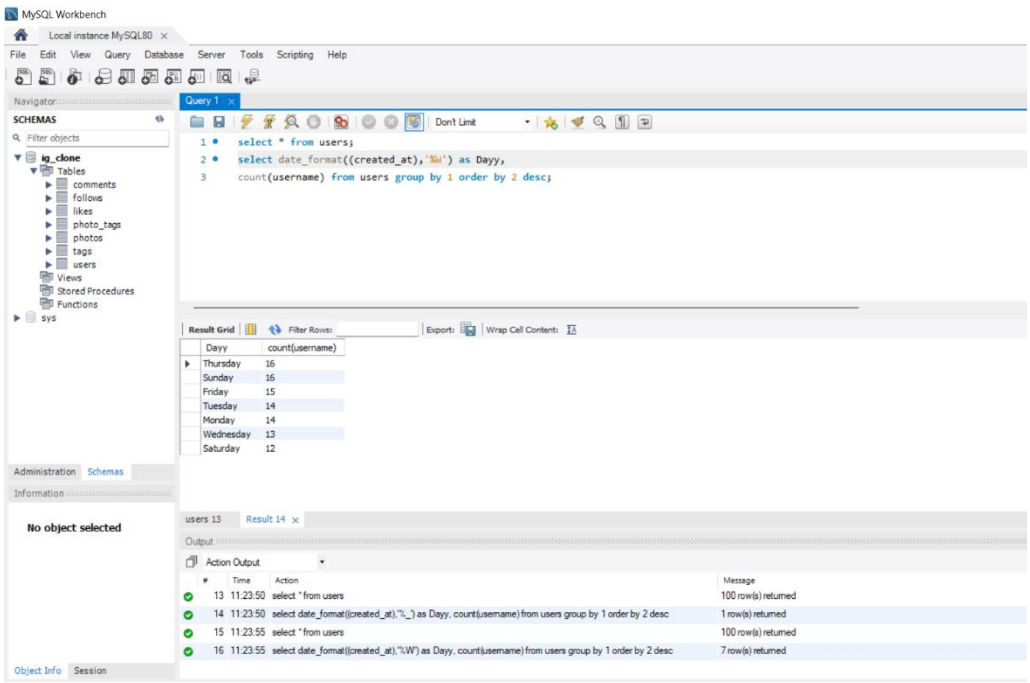
4. **Hashtag Research:** Identify and suggest the top five most commonly used hashtags on the platform.



Answer: In this, task we will need photo_tags and tags table from the database. using the SELECT query we will get the tag_name from **Tag** table and count the photo_id from **Photo_tags** table as (ht) **Hastag.** From the **photo_tag** (p is a short name for photo_tag) and we will inner join **tags** (t is the short name for tags).

The id of (t) is equal to the tag_id of (p) and group them by tag_name from (t) and arrange them in order by (ht) **hastag** in descending. Limit it up to 5 because we need only five tags.

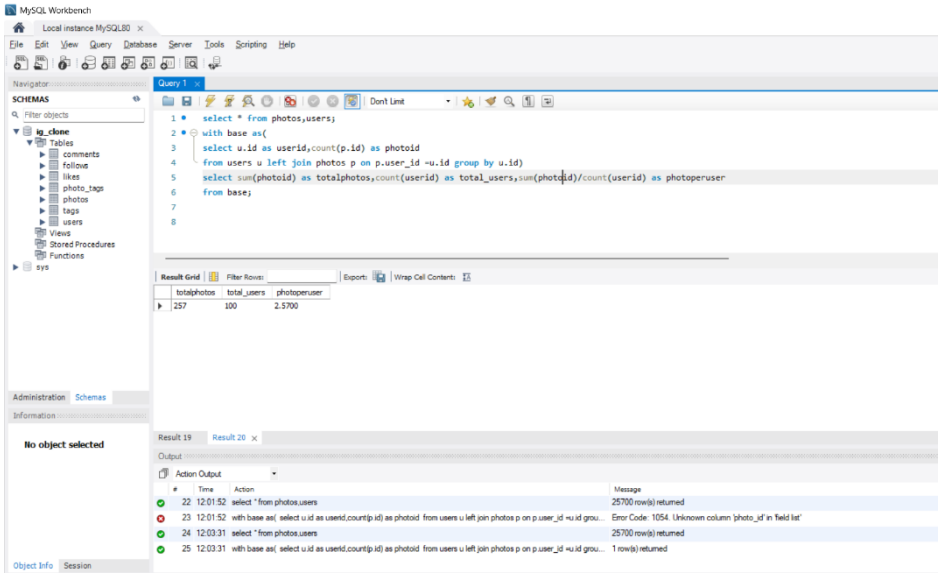
5. **Ad Campaign Launch:** Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.



Answer: In this, task we will need **USERS** table from the database. Using the Date_format (...,'%W') we Will convert the date of creation into day like (Monday, Tuesday...). And we will count the Usernames from the user table. Group all the data by the first column as **DAY** and arranging All the data order by username in Descending.

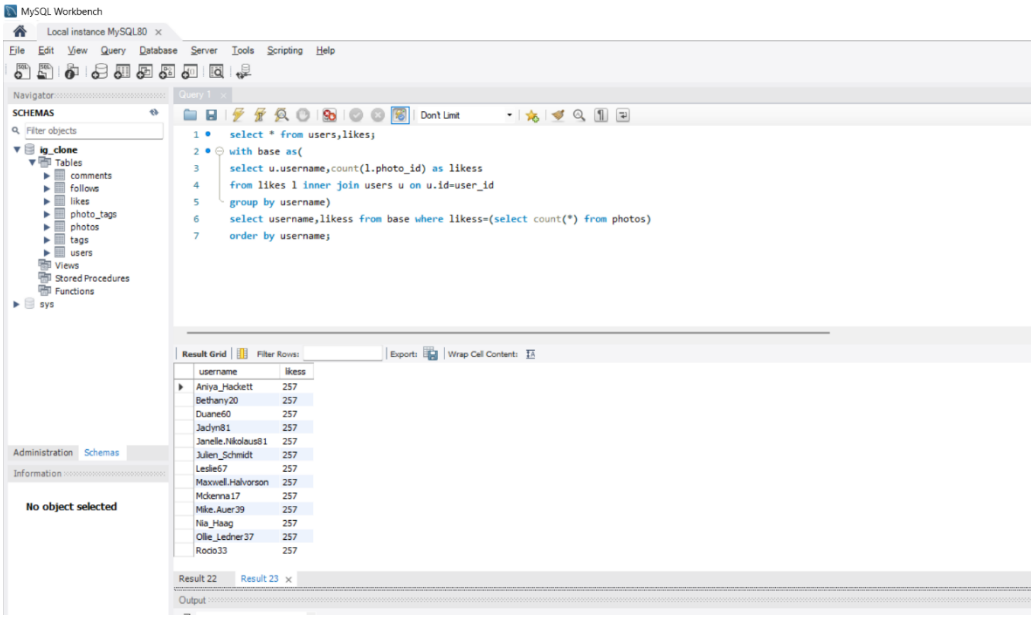
B) Investor Metrics:

1. **User Engagement**: 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.



Answer: In this, task we will need photos, users table from database. Then we will create an temporary table named as base. using select query **u.id** (user table's id column) renamed as userid and we will count the **p.id** (photo's id) and renamed as photoid from using **Left Join** we will join each user with there photos which will lead you to get every single users photos even with 0 photos and group them by user id.
then we will sum all the **photo id** and named as total photos and count the user id and named as total users then we will divide the total photos with the counted users and named as photo per user from the base.

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



Answer: In this, task we will use user and likes table from database. With creating an temporary table called **base** ,then we will get the username from **users** table and count the photo_id from table **likes** and name it as likess.

Later we will join the the id from user table and user_id from photos table. We will get the each likes done by a single user and group them by username and there likes.