**Project Description:**

Working as a Data Analyst in Meta for the Instagram branch, I have to extract useful insights from the dataset provided. These insights will be derived based on the user engagements with Instagram such as likes across photos, trendy tags, number of posts, period of highest usage, etc. These in turn will help Meta grow its Instagram platform to engage more users, support marketing platforms on Instagram, improve its user experience and ultimately help the business grow. The approach and the corresponding insights and results will be listed down in detail in the coming pages of this report.

**Tech-Stack Used:**

To implement this project, I have used the MySQL Workbench community version 8.0. There are two reasons for this:

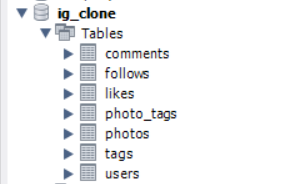
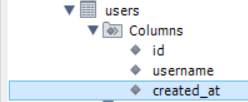
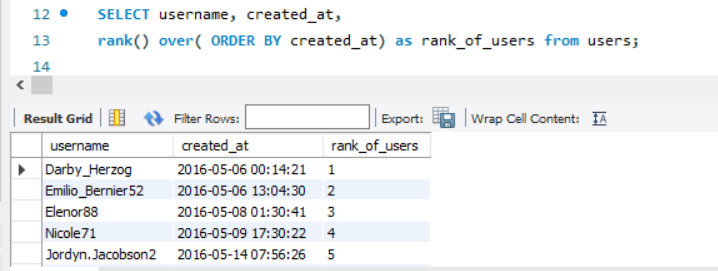
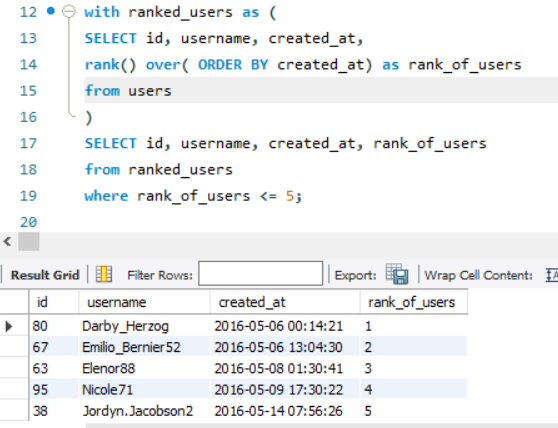
* MySQL’s community version allows news users to utilize their platform for analytics free of cost.
* I am familiar with the UI of this tool as I had practiced with the same tool in my college years.

Problem Description No. 1:

Loyal User Reward: The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

Task: Identify the five oldest users on Instagram from the provided database.

**Approach:**

1. From the dataset provided, we can see that there are in total **7 tables**:  
   
2. Now, to identify the five oldest **users** we can clearly look for our data in the **users table** and leave the rest.
3. When we look at the task, it says we need to identify **five oldest users** which means that we must rank all the users first. For this, we can use the SQL’s **rank()** function.
4. Next step is to identify on what basis should we rank them because if we look at the users table, we can see it has 3 columns. From these columns we can use the **created\_at** column to check the **date** when a user registered on Instagram.  
   
5. The most important step is that we don’t forget that using the rank function won’t automatically give us the **first five users** who registered on Instagram. Therefore, we must specify the **order** for **created\_at** column.
6. Finally, we can write our SQL query as shown below:  
   
7. In the above query, we are selecting username and the date at which the user created their account by selecting the created\_at column **from** the users table. Lastly, we are **creating** **a new column – rank\_of\_users** to find out the five oldest users based on the five oldest joining dates.
8. Here one can see that while writing **ORDER BY created\_at** we haven’t specified ASC or DESC because the default sorting order is ascending only – and we want that the column should be sorted from the oldest to the newest date. In this way the 1st rank will be given to the oldest user and the last rank to the latest user.
9. Now that we want only the first five users, we can use our query and nest it within another query using the CTE (Common Table Expression) as shown below:  
     
   We can create a **temporary table called “ranked\_users”** from “users” table and use this temporary table in our main query to extract the five oldest users.

**Insights:**

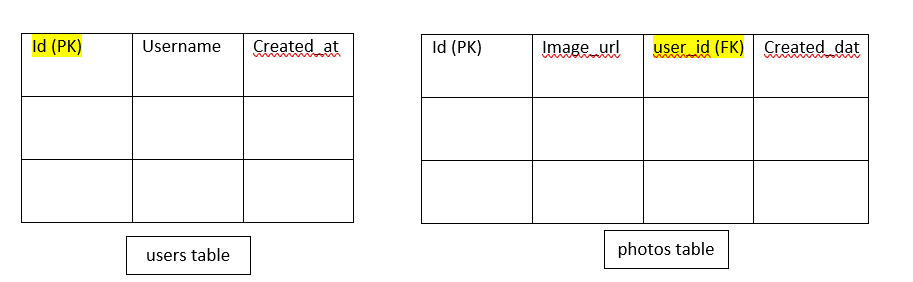
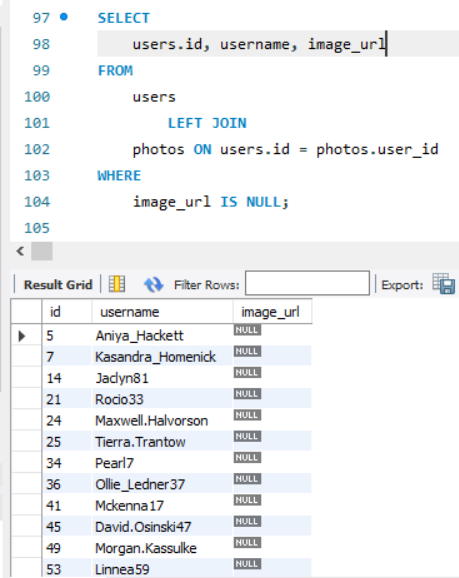
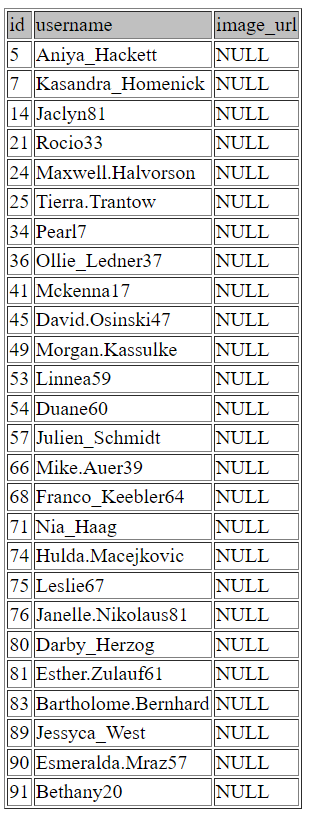
By rewarding the loyal users, other users will also get excited to start using Instagram and not exit from the platform. More users will register as a result helping in growing the business. Further Instagram can reward users who spend the most amount of time in a month or a week.

Problem Description No. 2:

Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.

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

**Approach:**

1. Our task involves two **keywords** – users and photo. From this we can identify that we will be required to analyze the **users** as well as the **photos** table provided to us in the dataset.
2. If we look at how the two tables – users and photos - are created, we will find that the **user\_id** column of the photos table acts as the **foreign key** to the users table, and it references the **id** column from it.  
   
3. Because we wish to identify the users who haven’t posted any photo, we will need the information from both the tables. This means that we must join the two tables **on** the condition that the **user\_id** of the photos table should be equal to the **id** of the users table.
4. To identify **the type of the join** we must understand from which table we want all the information. And it’s quite clear from our problem that we need to find **all the users** irrespective of whether or not they exist in the photos table. **This means we need all the records from the users table**. So we can use **left join** if we place users table on the left.
5. And to check whether or not a user has any image, we can look at the **image\_url** column **or** the **id** columns from the **photos table** and check if it’s **null**. For my query, I have used the image\_url column.
6. Finally, cumulating all this information, we can write our query as shown below:  
    

**Insights:**

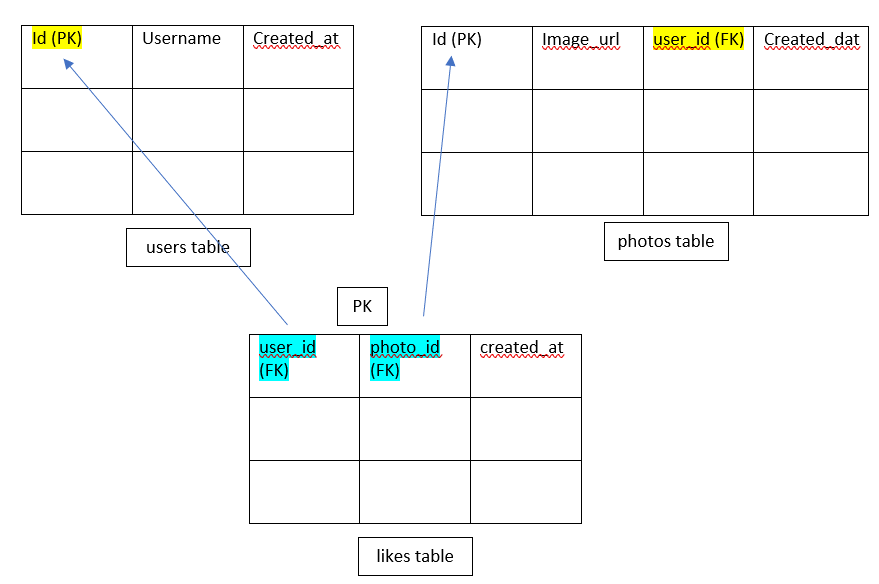
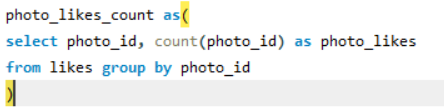
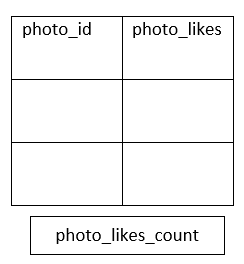
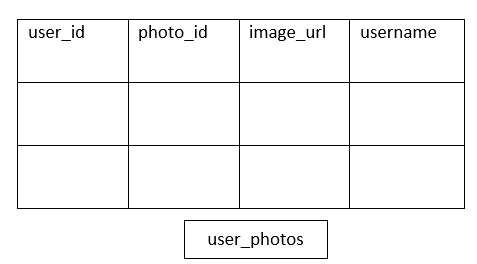
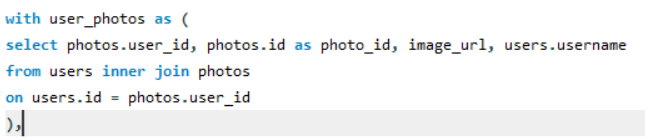
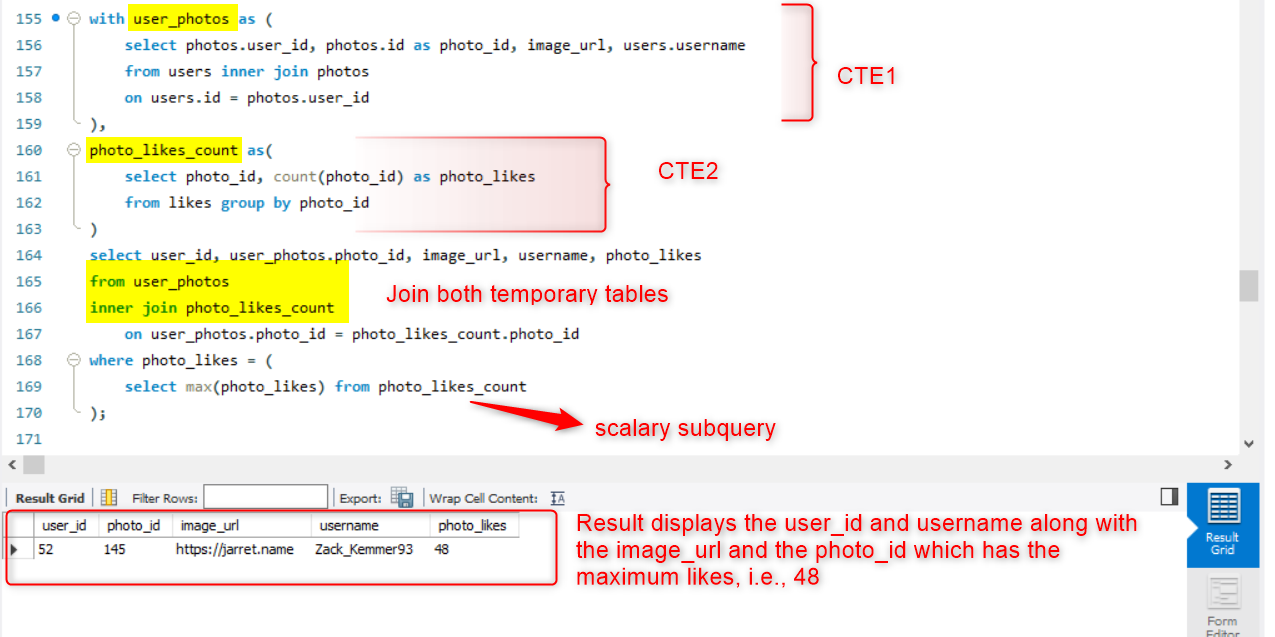
By identifying the users with 0 posts once Instagram sends promotional emails, the users will be motivated to join the platform and start posting images. These mails can talk about connecting with friends, sharing one’s opinions, using Instagram for marketing one’s personal products, getting connected with celebrities or one’s favorite sports person, etc. The images posted by the user can also help Instagram analyze what kind of things the user likes. For example – if a user posts images about travels, Instagram can suggest other travel accounts for the user to follow and hence keep them interested in the platform.

Problem Description No. 3:

Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.

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

**Approach:**

1. When we look at the problem description, we can see that now we have three keywords – user, likes and photo. From this we can conclude, we have to work on three tables from the dataset – **users, photos and likes.**In the likes table, the photo\_id column acts as the foreign key to the photos.id column and the user\_id column from the likes table acts as the foreign key to the users.id column.
2. First, we will work on the likes table to find out the number of likes of **each photo\_id** and create a **temporary table** so it can be used later to extract further information. This temporary table I have named as “photo\_likes\_count”.  
    
3. Now comes the important step, i.e., extracting the **user** information.  
   If we use this temporary table created at step 2 and join it with the photos table, we will **only** get the **user\_id** of the user who has got maximum number of likes on their photo. But since we need the complete user information, it will be easier to **first join the users and the photos table**. With this I have created another temporary table called “user\_photos”. Here I have selected the required columns, i.e., user\_id, photo\_id, image\_url and username.  
   
4. Finally, we can join these two temporary tables – **photo\_likes\_count and user\_photos**, with the filter that the **photo\_likes** should be the maximum value in that column.
5. Our combined query looks like:  
   

**Insights:**

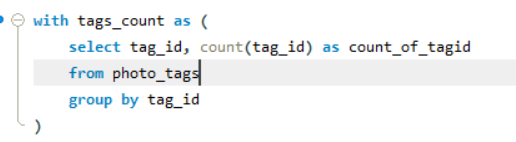
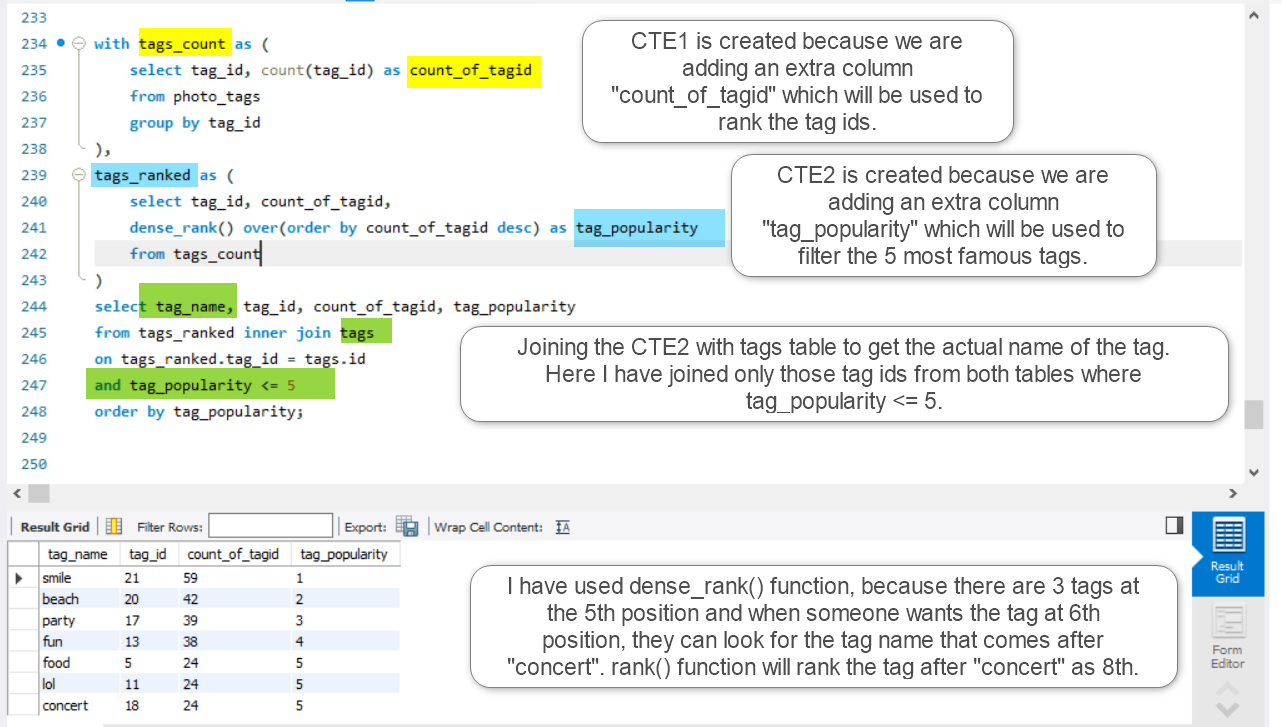
When Instagram awards a user who has got most likes on a photo, this user along with the other users on the platform will be encouraged to post more images and connect with more people so they can persuade them to like their images. With the help of such contests Instagram can increase engagement on its platform. More number of photos will mean more data for Instagram to know its users and with more data, we can derive more insights and help the business grow.

Problem Description No. 4:

Hashtag Research: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

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

**Approach:**

1. Looking at the problem, we can see that we have to work with the **tags** table. But even though there is no mention of the keyword **photos** here, it is obvious that the **tags** are used on **photos** only on the Instagram platform. This means we need two tables for this problem - tags and photo\_tags.
2. The first thing to do here is counting the number of times each tag has been used across all the photos on Instagram. I have created a **temporary table here called tags\_count** for this.  
   
3. Next, we must rank this tag\_id column based on the usage of the tags, i.e., nothing but ranking from the highest to lowest count of tags. In this part of the query, I have ordered the count\_of\_tagid column from the temporary table created in step 2 in descending order and created a new column to display the rank as tag\_popularity. The temporary table with this new tag\_popularity column, I have named as “tags\_ranked”.  
   
4. Because we need to identify what the actual tag is we must find out the tag name. For this, we must join the tags table with the temporary table tags\_ranked.
5. Finally, we will extract the 5 most popular tags by applying the filter to the temporary table “tags\_ranked”.

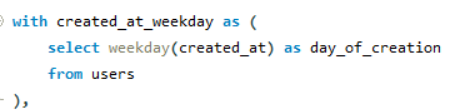
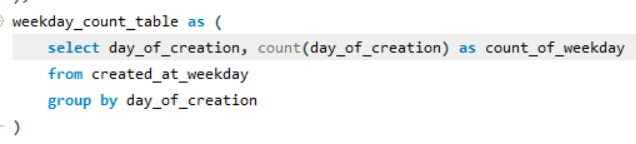
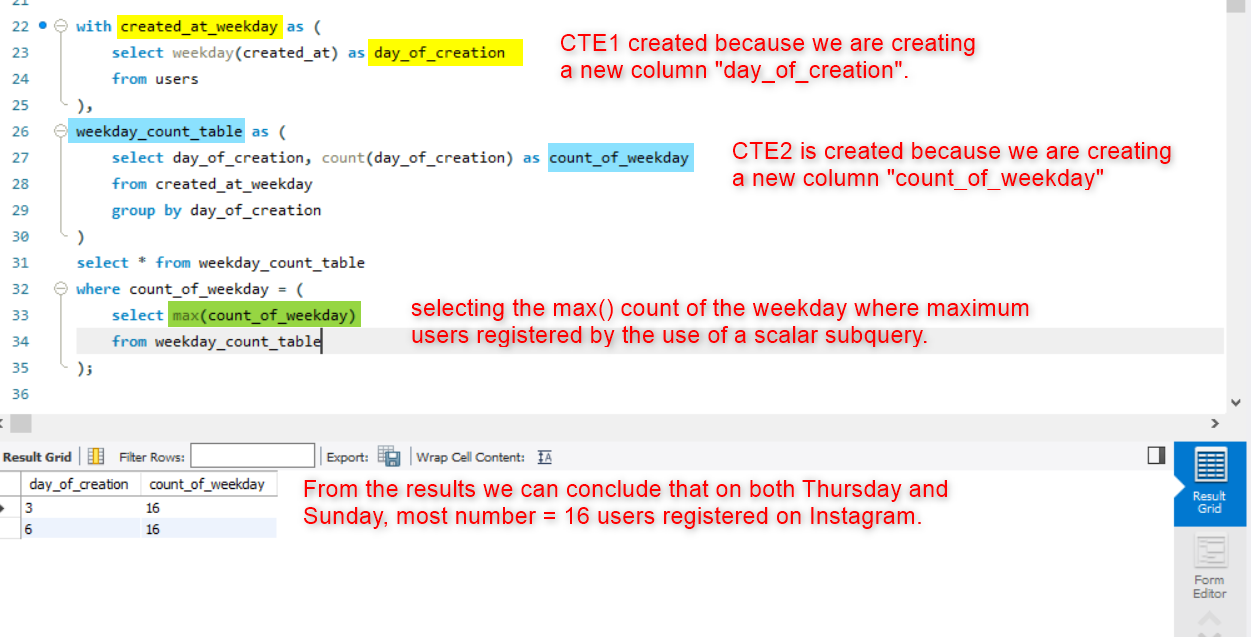
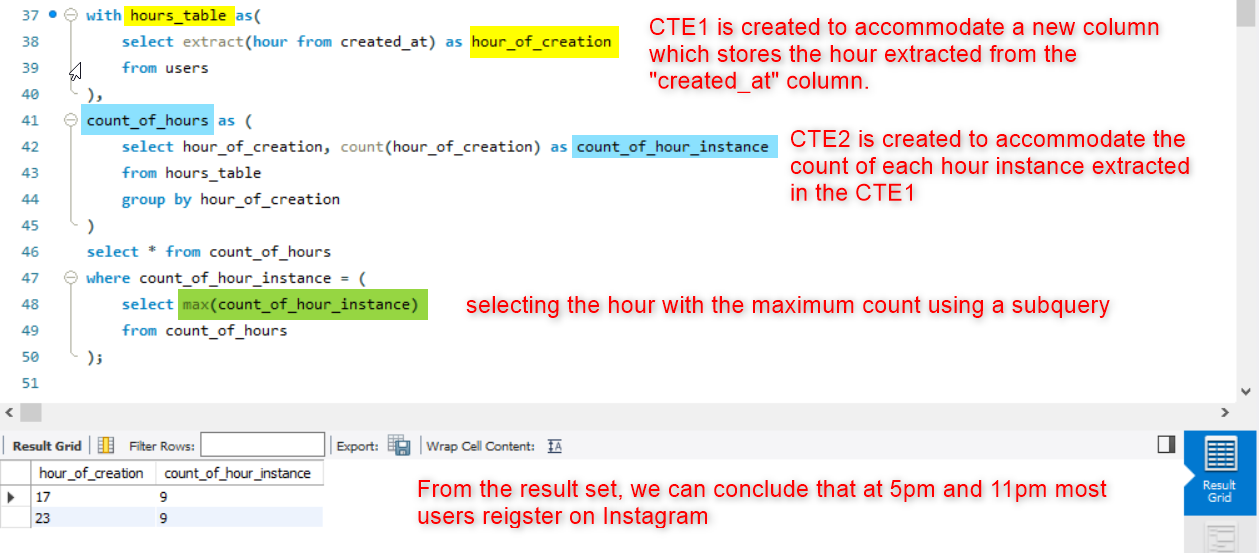
**Insights:**Users can follow particular tags to view the posts related to these tags. By using the popular tags, brands will be able to reach even those users who haven’t followed them. If we look at the result set, we can see that “smile” is the most popular tag. A brand selling power tools has no connection with ‘smile’ but using this tag in their post they will be able to reach users who have liked posts containing this tag and hence market their product to more users increasing their probability of selling the product.

Problem Description No. 5:

Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

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

**Approach:**

1. Table to analyze – users.
2. Identify the day of the week using the created\_at column from the users table. weekday(date) function allows us to do that. We will create a new column and in turn create a temporary table for this new column as shown below: This weekday() functions returns a number corresponding to each day of the week.  
   0 = Monday, 1 = Tuesday, 2 = Wednesday, 3 = Thursday, 4 = Friday, 5 = Saturday, 6 = Sunday.  
   
3. Next, count how many users registered on each day of the week using the new column created at step 2. Create another column to store this count and in turn another temporary table as shown below:  
   
4. From this temporary table “weekday\_count\_table” now we will select the day of the week which has the maximum count which is nothing but the day of the week when maximum users registered on Instagram.  
   
5. Similarly, we can get the time of the day when maximum users register on Instagram. Here we use the extract(hour from <date>) function.  
   

**Insights:**

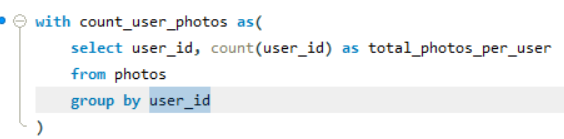
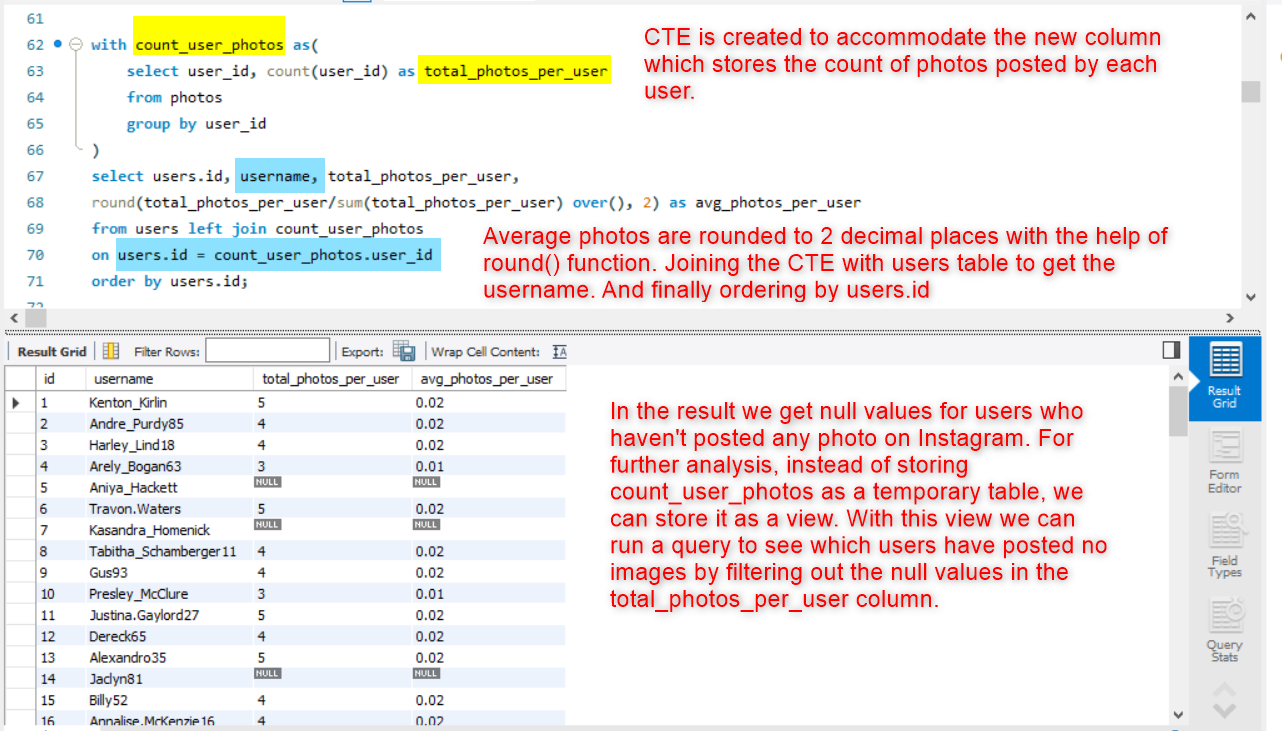
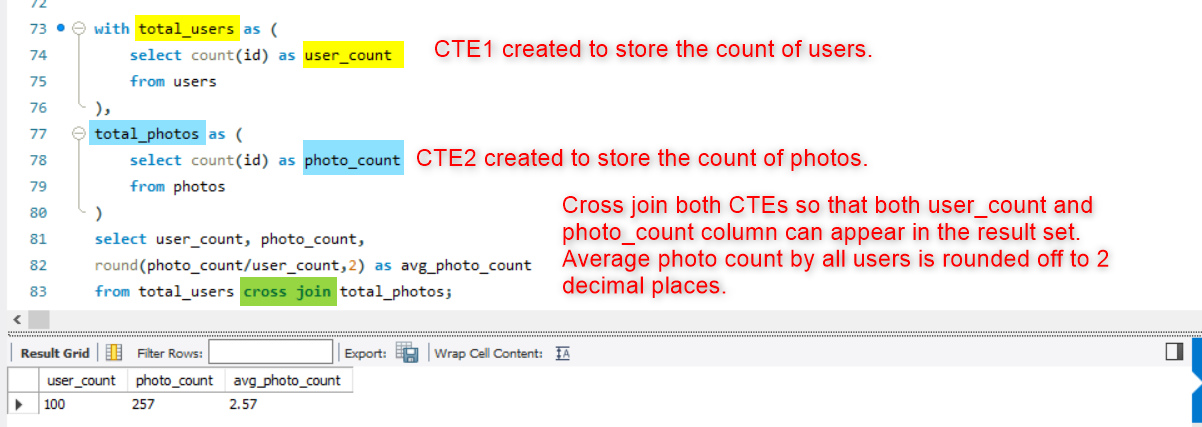
From the above two result sets we can conclude that the best day and time to launch an ad campaign is either on a Thursday or on a Sunday after 5 p.m. or after 11 p.m. This is important to note that when a user opens the Instagram platform they don’t usually scroll to the bottom of the feed and are more likely to engage with the posts shown at the top. If the ad is launched on a day or at a time before the user opens Instagram, it is less likely to be viewed much less be engaged with.

Problem Description No. 6:

User Engagement: Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

Task: 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.

**Approach:**

1. Tables to analyze – users and photos.
2. Count the total number of photos **each** user has posted from the **photos table** as shown below and create a temporary table with the new column.  
   
3. Next, we calculate the average photos posted by **each** user which is nothing but number of photos posted by 1 user which we have stored in the “total\_photos\_per\_user” column divided by the total photos across Instagram.
4. Total photos across Instagram can be calculated by the count() function by counting down the photos.id column or by using the sum() function down the “total\_photos\_per\_user” column. I have used the latter method.
5. Finally, to include the username we will join the two tables – users and count\_user\_photos on the user id condition as shown below.  
   
6. Total number of photos on Instagram divided by total number of users on Instagram can be queried as follows:  
   

**Insights:**

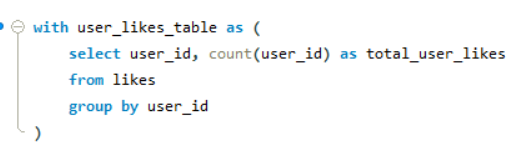
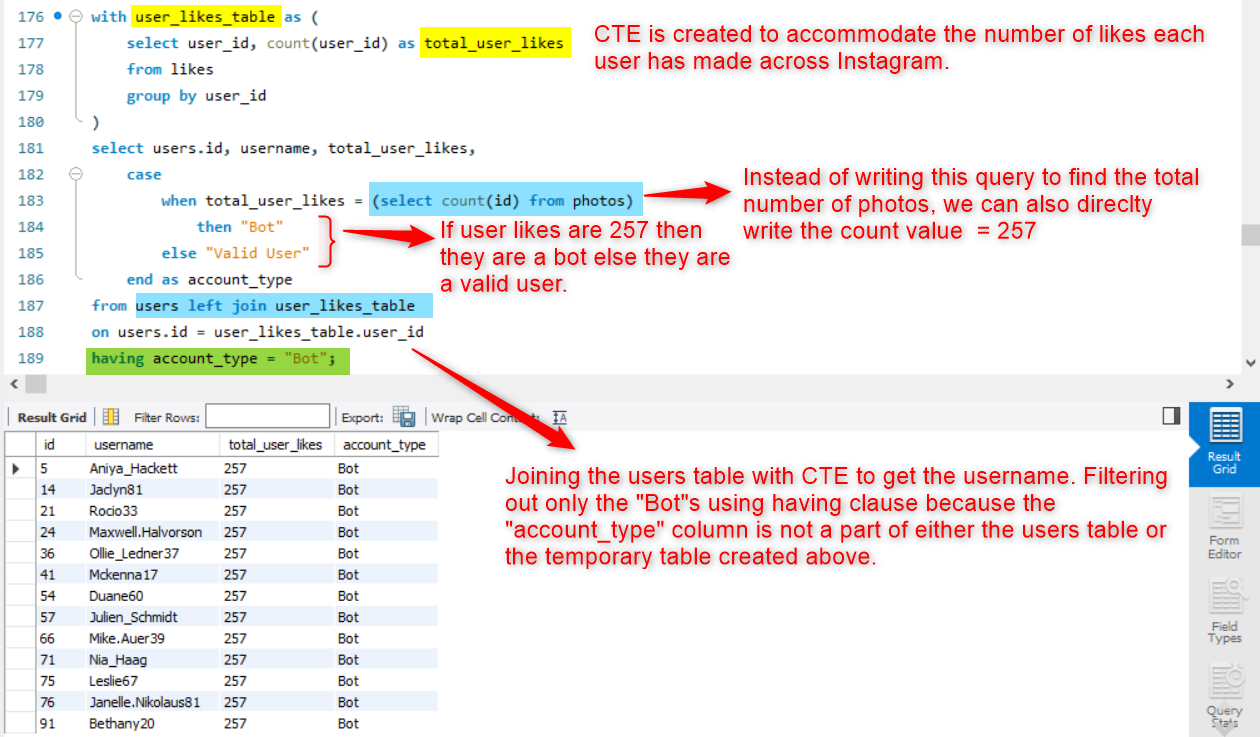
By looking at the average, we can see that it is not a good number if we want users to keep engaging with the platform. Both the averages – per user and total average of photos is better to decide the engagement than walking down the total\_photos\_per\_user column. Since average is mostly one number, we can see that no user has an average above 0.05. The total average is 2.57 while a good average would be above 50. Using this insight, marketing team can launch new campaigns to increase the average. This insight can also be matched with the pervious ones to launch contests which will encourage users to increase their personal average.

Problem Description No. 7:

Bots & Fake Accounts: Investors want to know if the platform is crowded with fake and dummy accounts.

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

**Approach:**

1. Tables to analyze – users and likes.
2. Find out the total number of likes **each** user has made from the **likes** table and create a temporary table to store this new column as shown below:  
   
3. Now we know if the total\_user\_likes is equal to the total number of photos posted across Instagram then that user is a bot. To get the information about that user we need to join the users table with the users\_likes\_table created in step 2. Using the case when clause as shown below, we can filter out the bots:  
   

**Insights:**

If a fake account is registered on Instagram, it will hamper the integrity of the platform. Similarly, bots will delude the investors that the business has potential while in truth most engagement is derived from these bots as we can see in the result set. This is why it is necessary to identify and remove them after which the correct business report can be generated which will help in gaining investors’ trust and help the business as well as the Instagram community grow.

**Overall Result:**

This project has increased my confidence in working with SQL queries especially with joins. A real-world practical application has helped me gain hands-on knowledge and experience on what my job would be as a Data Analyst. Apart from this, this particular project has helped me understand how Instagram works and as a user myself I now have an actual insight into things that I was using without any kind of awareness. The tags I follow, the ads I engage with, marketing team involvement, the posts I like, all these things are tracked in detail by the analytics team of Instagram which enhances my experiences as a user and also helps in growing the Instagram platform.