Project 2 : By Ramana Bansal

Instagram User Analytics

Project Description : The project focuses on a clone of Instagram user dataset. The data contains information regarding various activities such as user id, names, posts, likes, etc. of a common Instagram User. We will be using this data to derive insights such as oldest users, inactive users, bots, average number of posts, common hashtags used. The data will also be used to find the most liked photo and the most popular day of registration which will aid in finding contest winners and launching ad campaigns.

Approach: The data was grabbed and analysed using MySql queries. MySql Workbench was used to run various queries and observe the outputs.

Tech-Stack Used: MySQL Workbench 8.0.33 : To Execute the Sql queries.

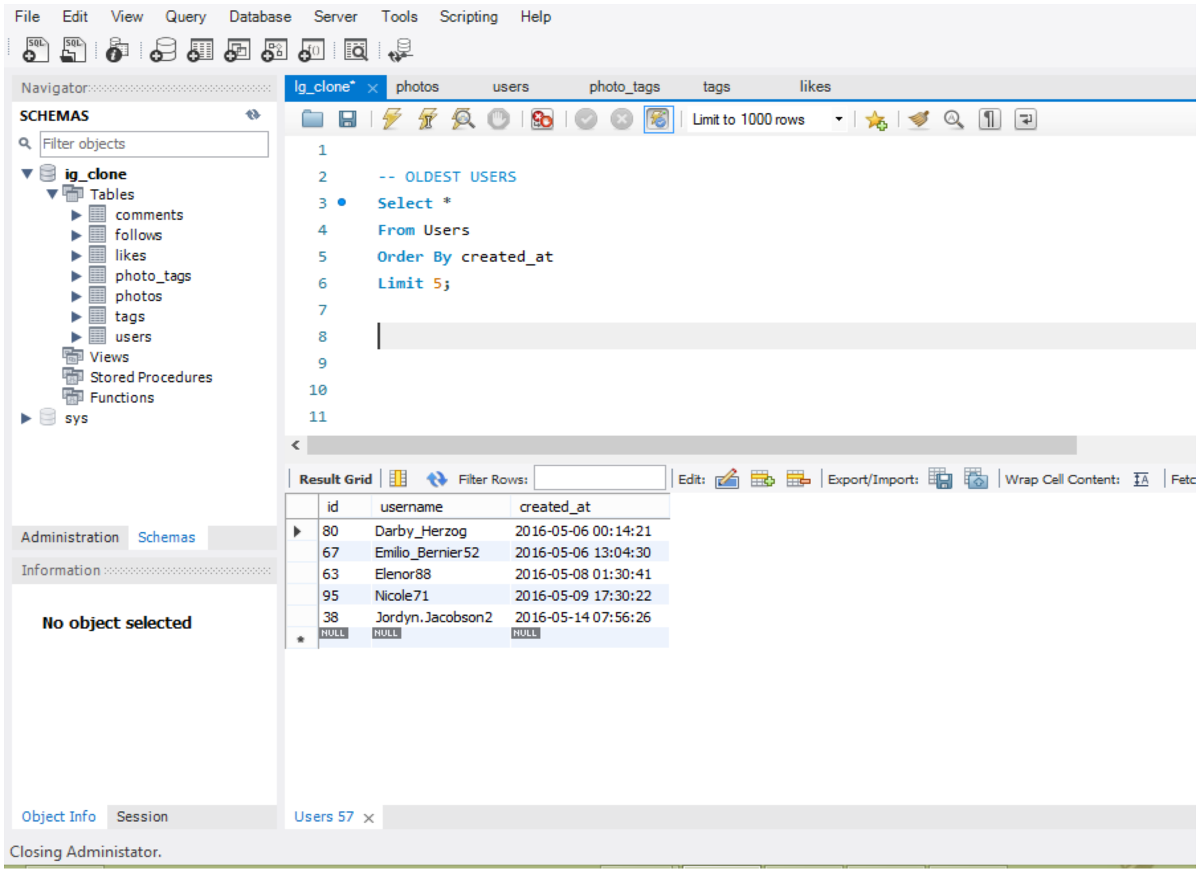
Insights: The project was helpful in getting a general idea about the practical application of data analysis. It also helped in understanding some of the data-driven processes behind various social media such as Instagram.

Result: The project made me feel a little more confident with MySql and its use in data cleaning and interaction. It also gave insights regarding how data can be used to make decisions and derive policies for businesses.

A) Marketing:

1. **Rewarding Most Loyal Users:** Find the 5 oldest users of the Instagram from the database provided.

   Select *
   From Users
   Order By created_at
   Limit 5;



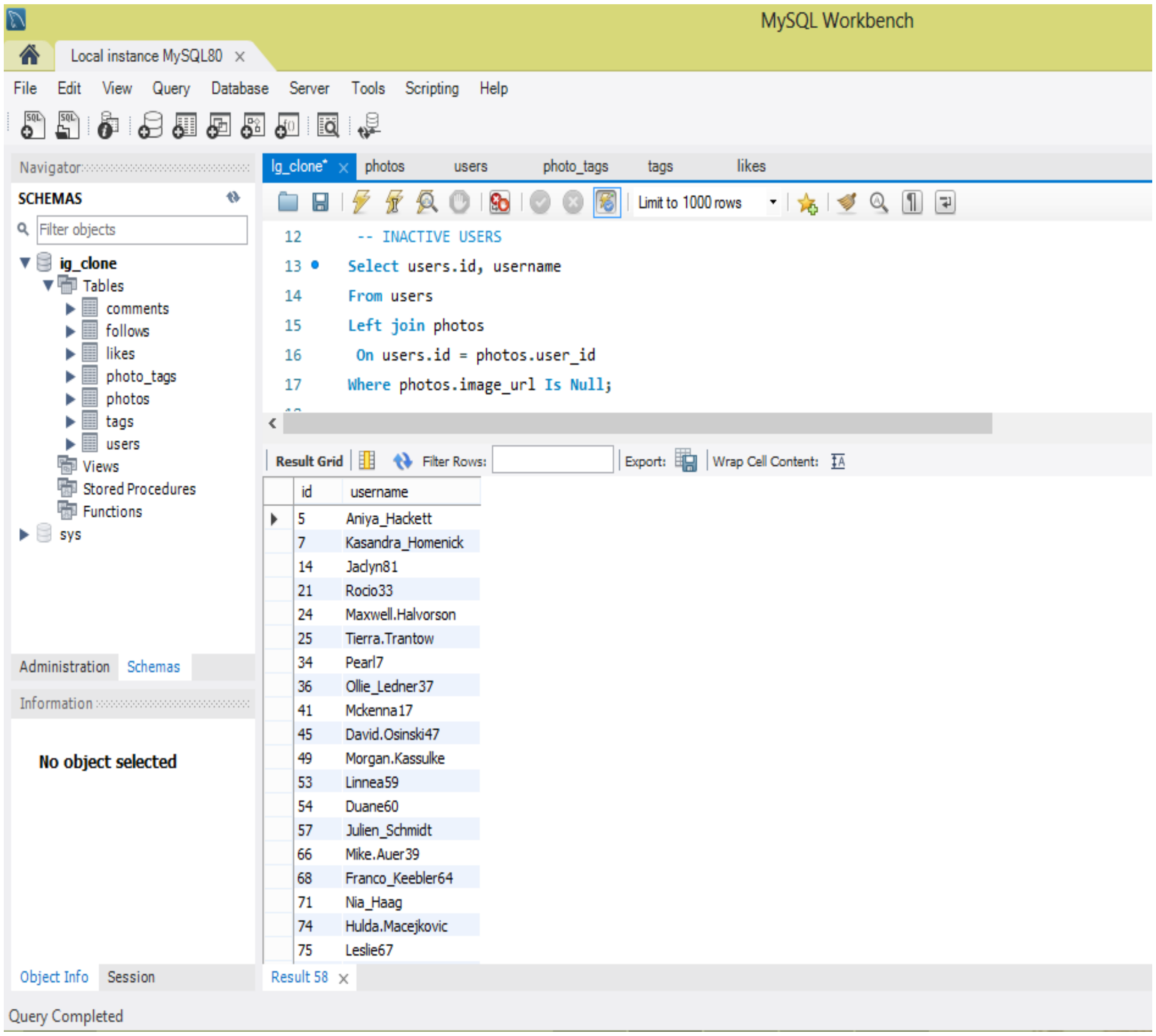
```
-- OLDEST USERS
Select *
From Users
Order By created_at
Limit 5;
```

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
NULL	NULL	NULL

We sort the users table at created_at table in descending order, and set the limit to 5 to get the five oldest users. The created_at table gives the timestamp for when the account was created.

2. **Remind Inactive Users to Start Posting:** Find the users who have never posted a single photo on Instagram.

   Select users.id, username
   From users
   Left join photos
     On users.id = photos.user_id
   Where photos.image_url Is Null;



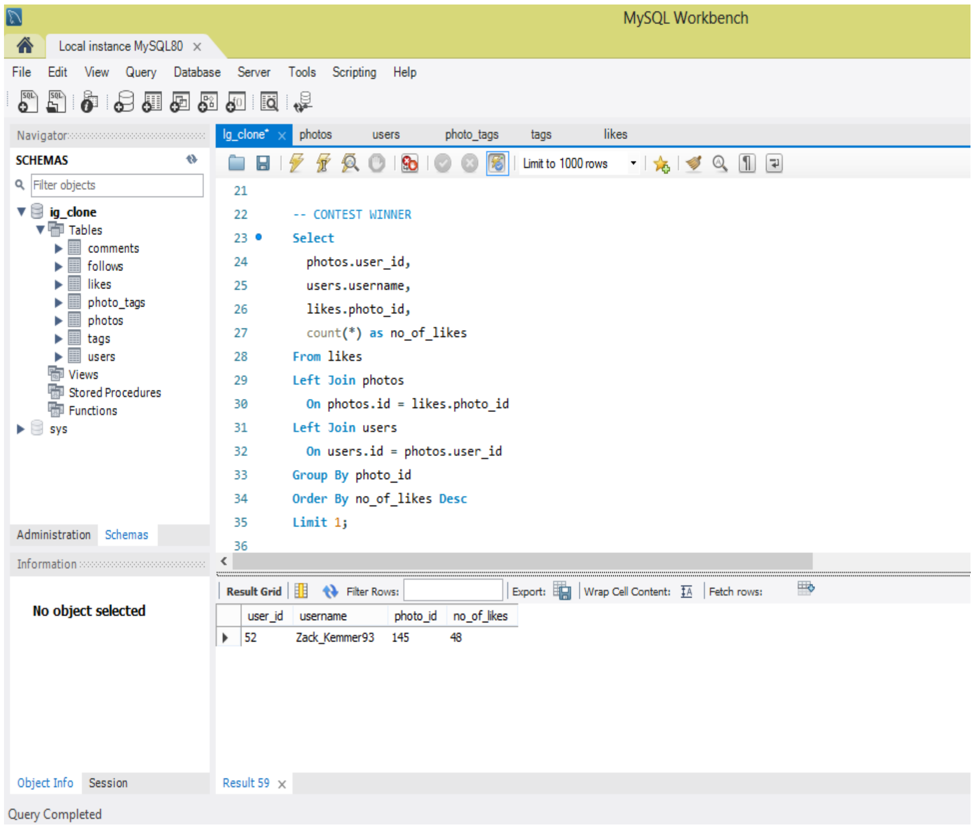
```
-- INACTIVE USERS
Select users.id, username
From users
Left join photos
 On users.id = photos.user_id
Where photos.image_url Is Null;
```

id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67

We used left join to join the tables users and photos, and extracted the usernames which had null values for image_urls.

3. **Declaring Contest Winner:** Identify the winner of the contest and provide their details to the team.

```
Select
   photos.user_id,
   users.username,
   likes.photo_id,
   count(*) as no_of_likes
 From likes
 Left Join photos
   On photos.id = likes.photo_id
 Left Join users
   On users.id = photos.user_id
 Group By photo_id
 Order By no_of_likes Desc
 Limit 1;
```



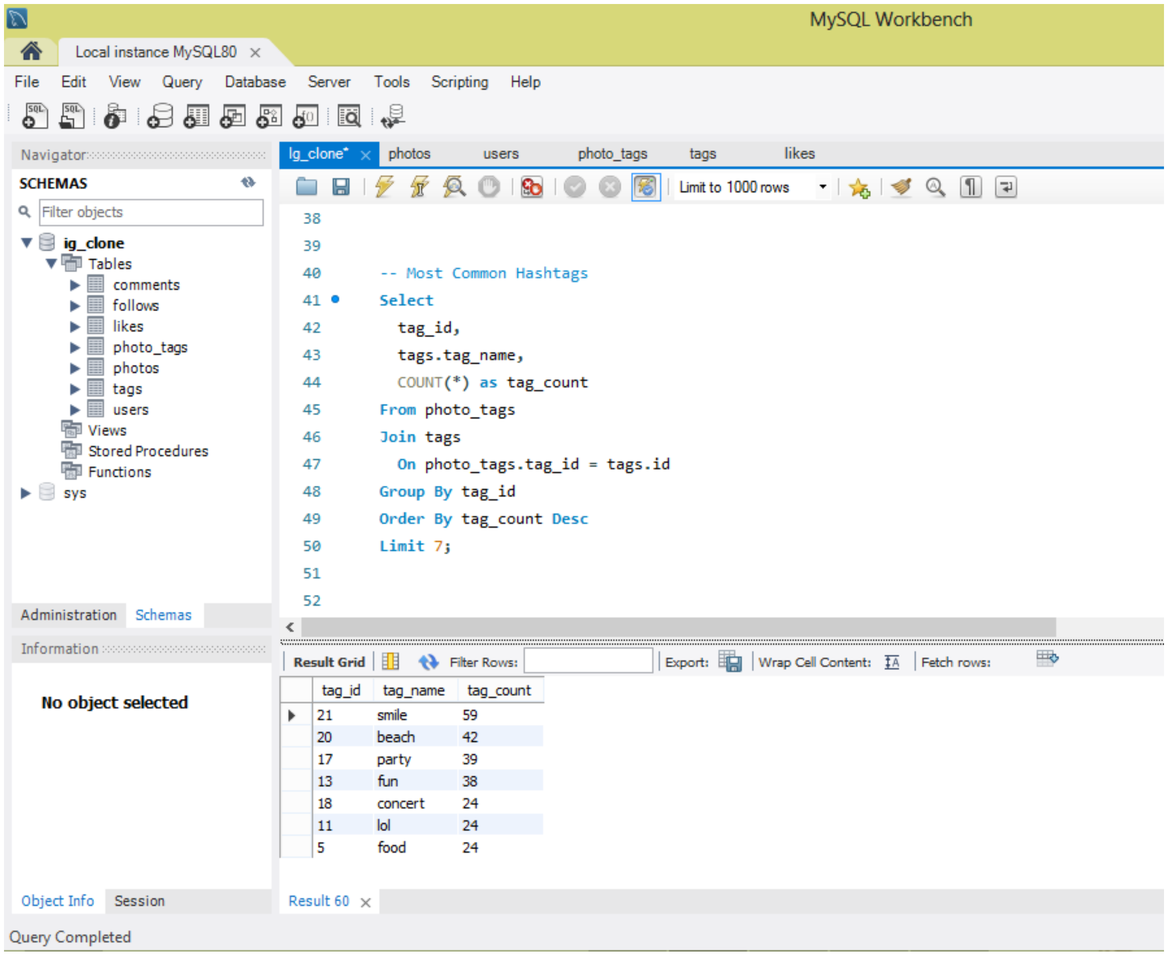
```
-- CONTEST WINNER
Select
  photos.user_id,
  users.username,
  likes.photo_id,
  count(*) as no_of_likes
From likes
Left Join photos
  On photos.id = likes.photo_id
Left Join users
  On users.id = photos.user_id
Group By photo_id
Order By no_of_likes Desc
Limit 1;
```

user_id	username	photo_id	no_of_likes
52	Zack_Kemmer93	145	48

We have joined the tables photos, likes and users. We found the number of likes each photo received using Group By and Count for photo_id.

4. **Hashtag Researching:** Identify and suggest the top 5 most commonly used hashtags on the platform.

```
Select
   tag_id,
   tags.tag_name,
   COUNT(*) as tag_count
 From photo_tags
 Join tags
   On photo_tags.tag_id = tags.id
 Group By tag_id
 Order By tag_count Desc
 Limit 7;
```



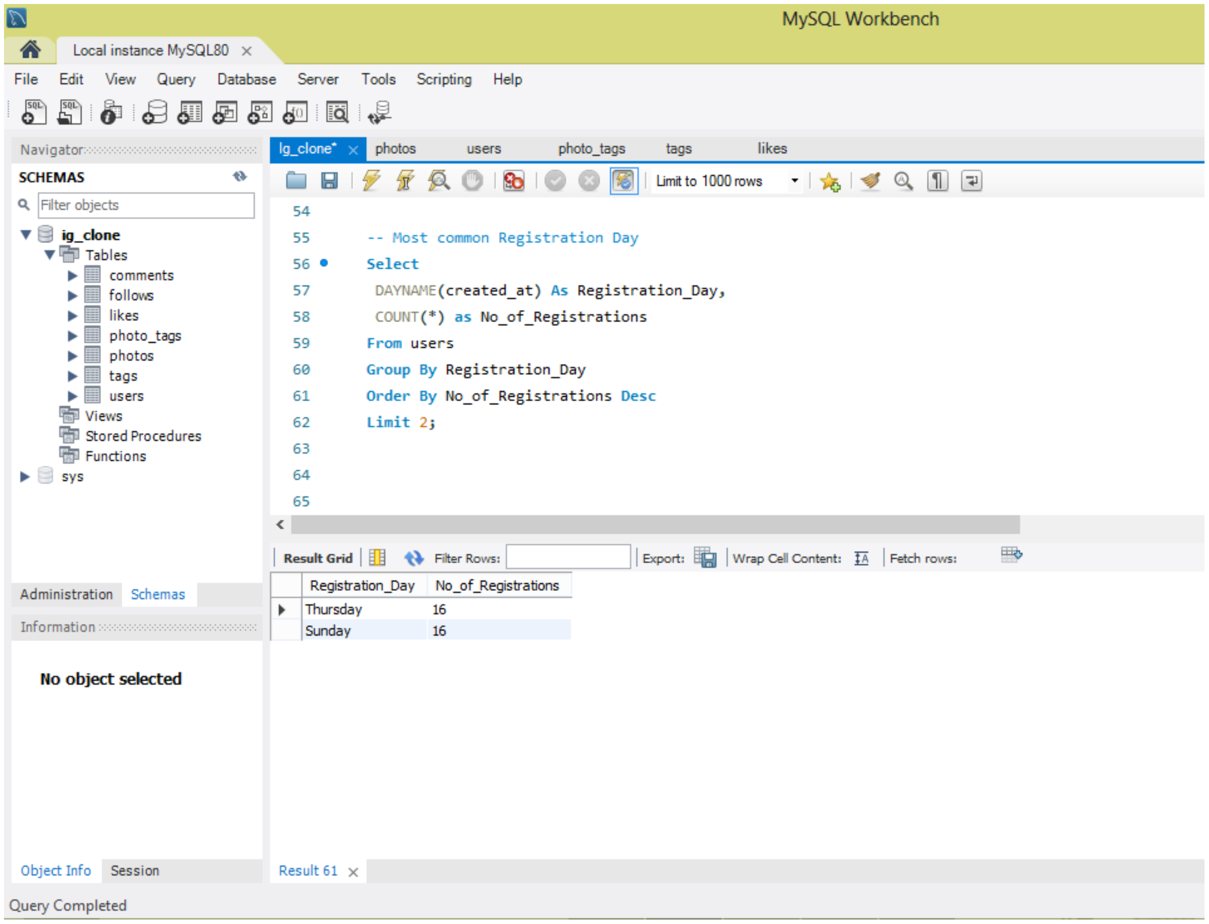
```
-- Most Common Hashtags
Select
  tag_id,
  tags.tag_name,
  COUNT(*) as tag_count
From photo_tags
Join tags
  On photo_tags.tag_id = tags.id
Group By tag_id
Order By tag_count Desc
Limit 7;
```

tag_id	tag_name	tag_count
21	smile	59
20	beach	42
17	party	39
13	fun	38
18	concert	24
11	lol	24
5	food	24

We used Group By and Count for tag_id to count the number of times a tag was used. The tables photo_tags and tags were joined to get the names of tags.The 5th position is shared by three hashtags, therefore we have set the limit to 7.

5. **Launch AD Campaign:** What day of the week do most users register on? Provide insights on when to schedule an ad campaign.

```
Select
  DAYNAME(created_at) As Registration_Day,
  COUNT(*) as No_of_Registrations
 From users
 Group By Registration_Day
 Order By No_of_Registrations Desc
 Limit 2;
```



```
-- Most common Registration Day
Select
 DAYNAME(created_at) As Registration_Day,
 COUNT(*) as No_of_Registrations
From users
Group By Registration_Day
Order By No_of_Registrations Desc
Limit 2;
```

Registration_Day	No_of_Registrations
Thursday	16
Sunday	16

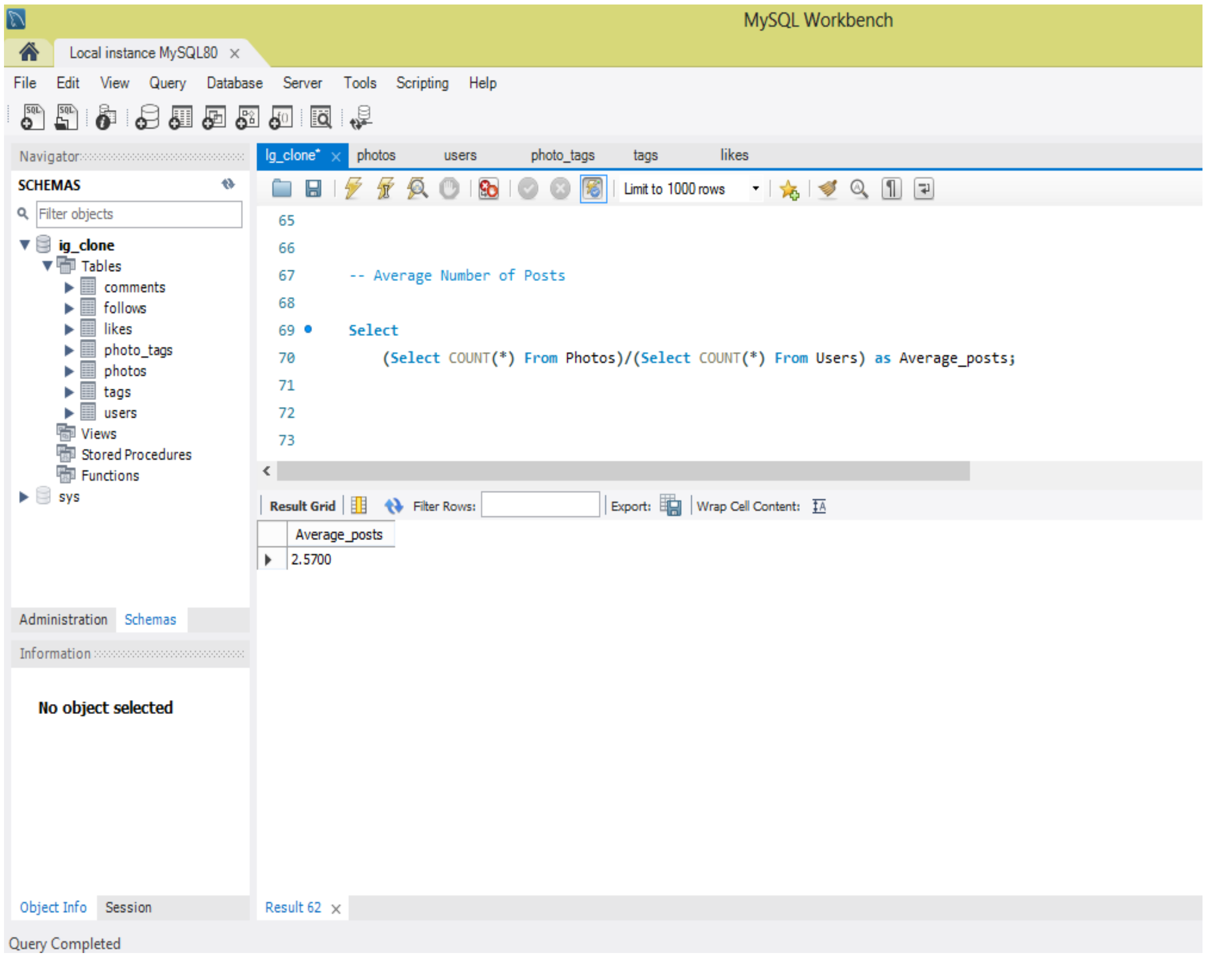
Query Completed

Both Thursday and Sunday have the maximum number of registrations, therefore we have set the limit to 2.

B) Investor Metrics:

User Engagement: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

```
Select
    (Select COUNT(*) From Photos)/(Select COUNT(*) From Users) as
Average_posts;
```



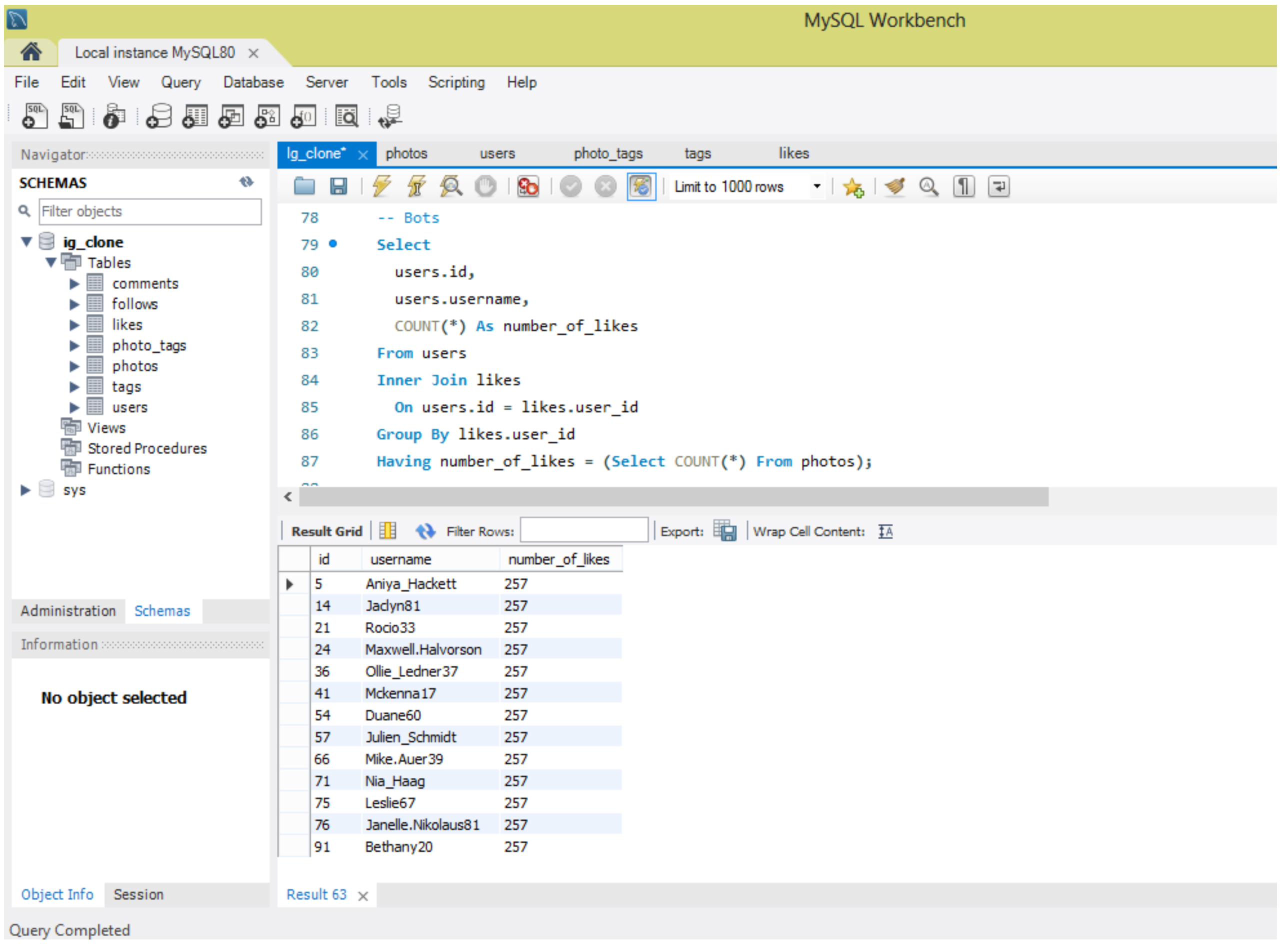
```
-- Average Number of Posts

Select
    (Select COUNT(*) From Photos)/(Select COUNT(*) From Users) as Average_posts;
```

```
Average_posts
2.5700
```

Subquery was used to find the average number of posts. The average number of posts for this dataset is 2.57.

Bots & Fake Accounts: 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).

```
Select
   users.id,
   users.username,
   COUNT(*) As number_of_likes
 From users
 Inner Join likes
   On users.id = likes.user_id
 Group By likes.user_id
 Having number_of_likes = (Select COUNT(*) From photos);
```



```
-- Bots
Select
  users.id,
  users.username,
  COUNT(*) As number_of_likes
From users
Inner Join likes
  On users.id = likes.user_id
Group By likes.user_id
Having number_of_likes = (Select COUNT(*) From photos);
```

id	username	number_of_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

The number of likes for each user_id was counted and compared to the total number of photos. The tables users and likes were joined to retrieve user information for the bots that had liked all the photos.