

Instagram User Analytics

05.03.2023

Aayushi Singh

Overview

Instagram user analytics is used to identify how the user engage with the concern digital project and based on the needs of the digital, product and marketing team we perform analysis on the dataset abstracted through the interaction of the user on the product

Approach

- The database is uploaded on the MySQL workbench to make it accessible and it is much ebay to use and handle data
- Through the Data Modeling and by the process of Reverse Engineering create an EER model of your instagram database it help you to understand how all the tables are connected and primary key with respect to the tables
- As as per the statement need join the two tables, the EER model helps you to understand through the lines and network which is the foreign key to the table you need to join and by click in on the table you can also understand each and every element of a column can now join to tables then by the sql you can derived the necessary results

Tech-Stack Used

Hardware

My SQL Workbench

Software

MySQL

Instagram database:

https://docs.google.com/document/d/1-WhNRX1iYJlz7e5l28DMPWgsPkIpE_w6/edit

Tasks

Based on the database provided , the table in the instagram database are

Users

Comment

Photos

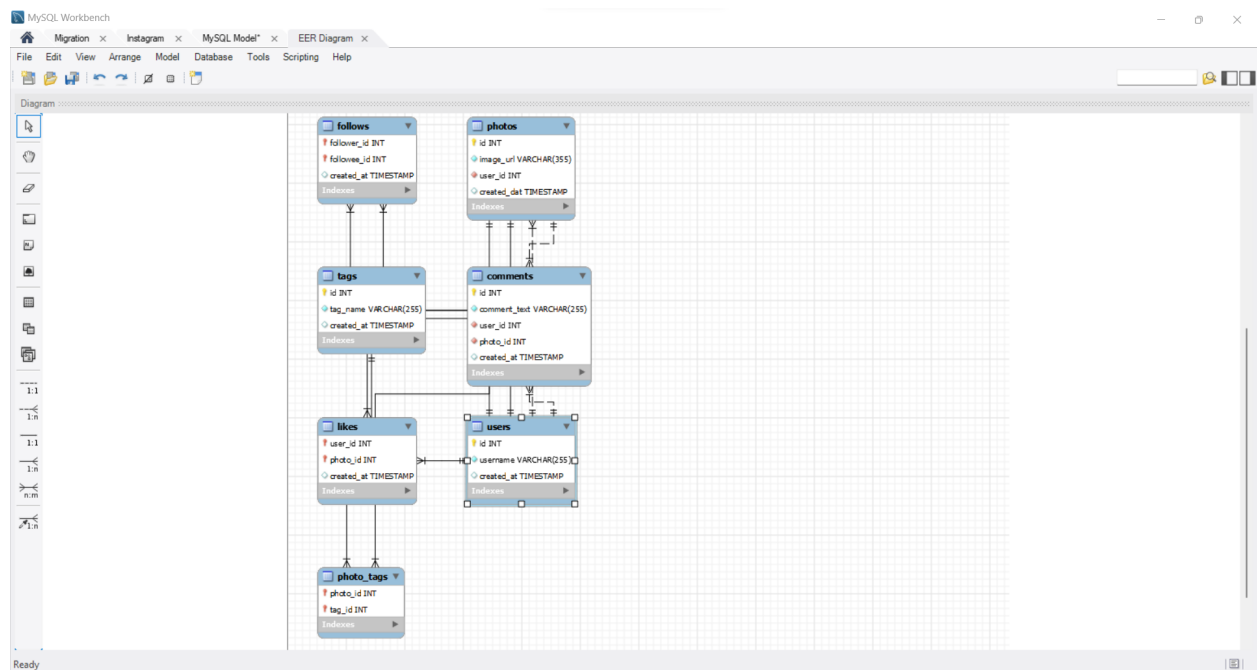
Follows

Tag

Likes

Photo_tags

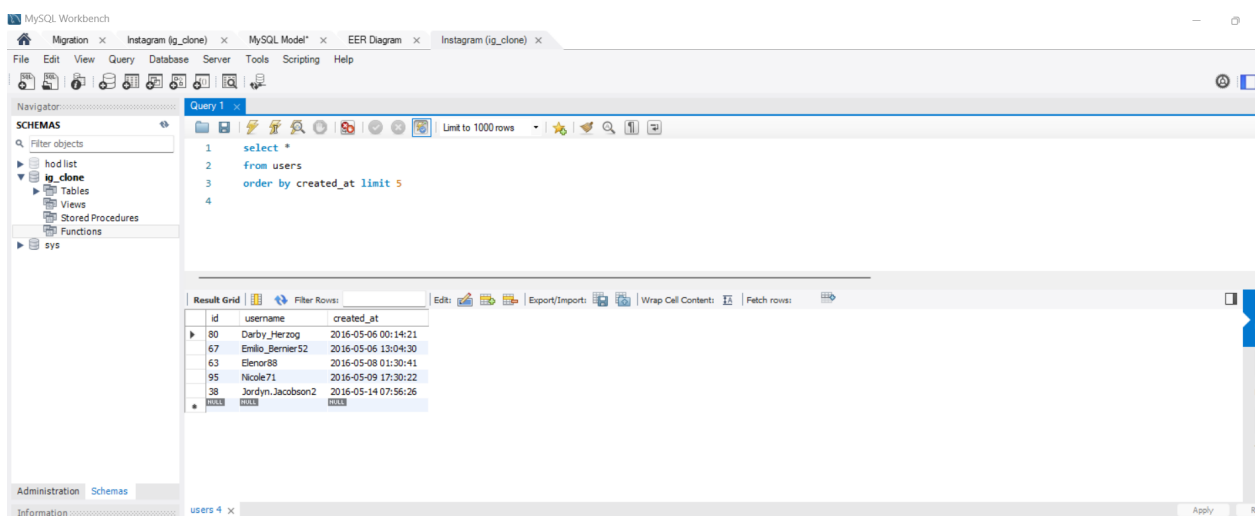
On the basis of the following EER Model



MARKETING

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

Your Task: Find the 5 oldest users of the Instagram from the database provided



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

The screenshot shows the MySQL Workbench interface with the following components:

- Navigator:** Shows the database structure with 'ig_clone' selected.
- Query Editor:** Contains the following SQL query:


```

      1 select *
      2 -- username
      3 from users
      4 left join photos
      5 on photos.user_id = users.id
      6 -- where photos.id IS NULL
      7 where image_url IS NULL
      8
      9
      10
      11
      
```
- Result Grid:** Displays the results of the query. The first table shows columns: id, username, created_at, id, image_url, user_id, created_at. The second table shows the same columns with data for users who have never posted a photo.

id	username	created_at	id	image_url	user_id	created_at
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kassandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jacyln81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
21	Rocio33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
24	Maxwell_Halvorson	2017-04-18 02:32:44	NULL	NULL	NULL	NULL
25	Tierra_Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	Mckenna17	2016-07-17 17:26:45	NULL	NULL	NULL	NULL
45	David_Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
49	Morgan_Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL
54	Duane60	2016-12-21 04:43:38	NULL	NULL	NULL	NULL
57	Julien_Schmidt	2017-02-02 23:12:48	NULL	NULL	NULL	NULL
66	Mike_Auer39	2016-07-01 17:36:15	NULL	NULL	NULL	NULL
68	Franco_Keebler64	2016-11-13 20:09:27	NULL	NULL	NULL	NULL
71	Nia_Haag	2016-05-14 15:38:50	NULL	NULL	NULL	NULL
74	Hilda_Macejkovic	2017-01-25 17:17:28	NULL	NULL	NULL	NULL
...

Insight :By Using the “ where image_url IS NULL “ We can filter out all the users that have never posted a single photo on Instagram

The screenshot shows the MySQL Workbench interface with the following components:

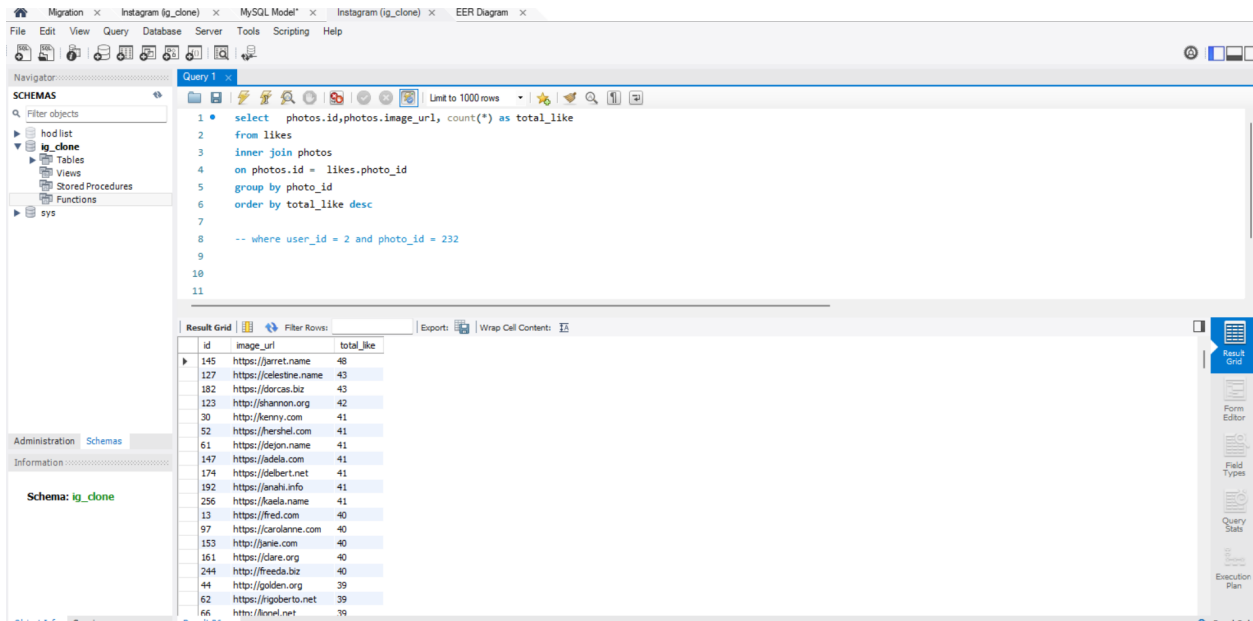
- Navigator:** Shows the database structure with 'ig_clone' selected.
- Query Editor:** Contains the following SQL query:


```

      1 select username
      2 from users
      3 left join photos
      4 on photos.user_id = users.id
      5 where image_url IS NULL
      6
      7
      8
      9
      
```
- Result Grid:** Displays the results of the query, showing a list of usernames:

username
Kassandra_Homenick
Jacyln81
Rocio33
Maxwell_Halvorson
Tierra_Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David_Osinski47
Morgan_Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike_Auer39
Franco_Keebler64
Nia_Haag
Hilda_Macejkovic
Leslie67
Tamara_Kuhlman

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



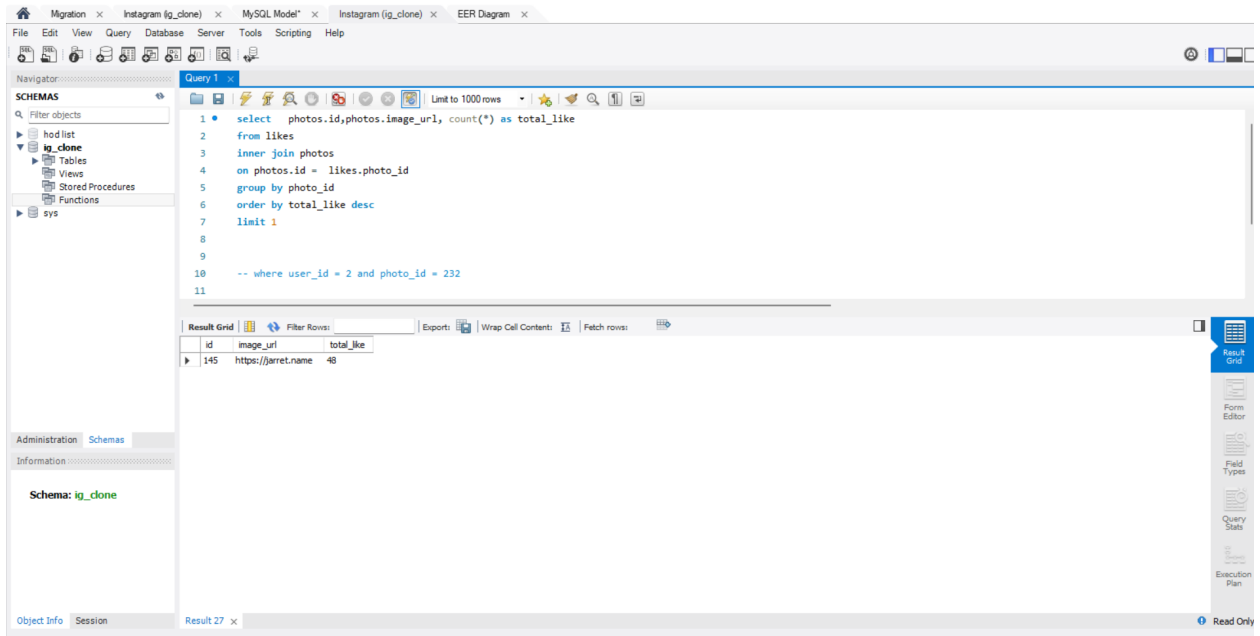
The screenshot shows a database management tool interface with a SQL query editor and a result grid. The query is as follows:

```
1 select photos.id, photos.image_url, count(*) as total_like
2 from likes
3 inner join photos
4 on photos.id = likes.photo_id
5 group by photo_id
6 order by total_like desc
7
8 -- where user_id = 2 and photo_id = 232
9
10
11
```

The result grid displays the following data:

id	image_url	total_like
145	https://barret.name	48
127	https://celestine.name	43
182	https://dorcias.biz	43
123	http://shannon.org	42
30	http://kenny.com	41
52	https://hershel.com	41
61	https://degon.name	41
147	https://jedela.com	41
174	https://delbert.net	41
192	https://anahi.info	41
256	https://kaela.name	41
13	https://fred.com	40
97	https://carolanne.com	40
153	http://janie.com	40
161	https://clare.org	40
244	http://freeda.biz	40
44	http://golden.org	39
62	https://rigoberto.net	39
66	http://lonnel.net	39

Insight :Inner join will remove all the users that do not have any photos and by using “limit 1” we can take the top most name with most likes as the data is sorted in descending order



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

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```

1 select tag_name, tag_id, count(*) as mosttag
2 from tags
3 inner join photo_tags
4 on tags.id=photo_tags.tag_id
5 group by tag_name
6 order by mosttag desc

```

The 'Result Grid' shows the results of the query, sorted by the number of times each tag is used in descending order. The results are as follows:

tag_name	tag_id	mosttag
smile	21	59
beach	20	42
party	17	39
fun	13	38
concert	18	24
food	5	24
lol	11	24
hair	15	23
happy	12	22
beauty	8	20
dreamy	10	20
drunk	19	19
fashion	16	19
sunset	1	19
landscape	4	17
style	14	17
sunrise	3	17
photogra...	2	16
stunning	9	16

Insight : There is the list of all the hashtags that are used and on how many times they are used in descending order, by putting of “limit 5” we can filter out the top 5 hashtags we need to use to analysis

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```

1 select tag_name, tag_id, count(*) as mosttag
2 from tags
3 inner join photo_tags
4 on tags.id=photo_tags.tag_id
5 group by tag_name
6 order by mosttag desc
7 limit 5;

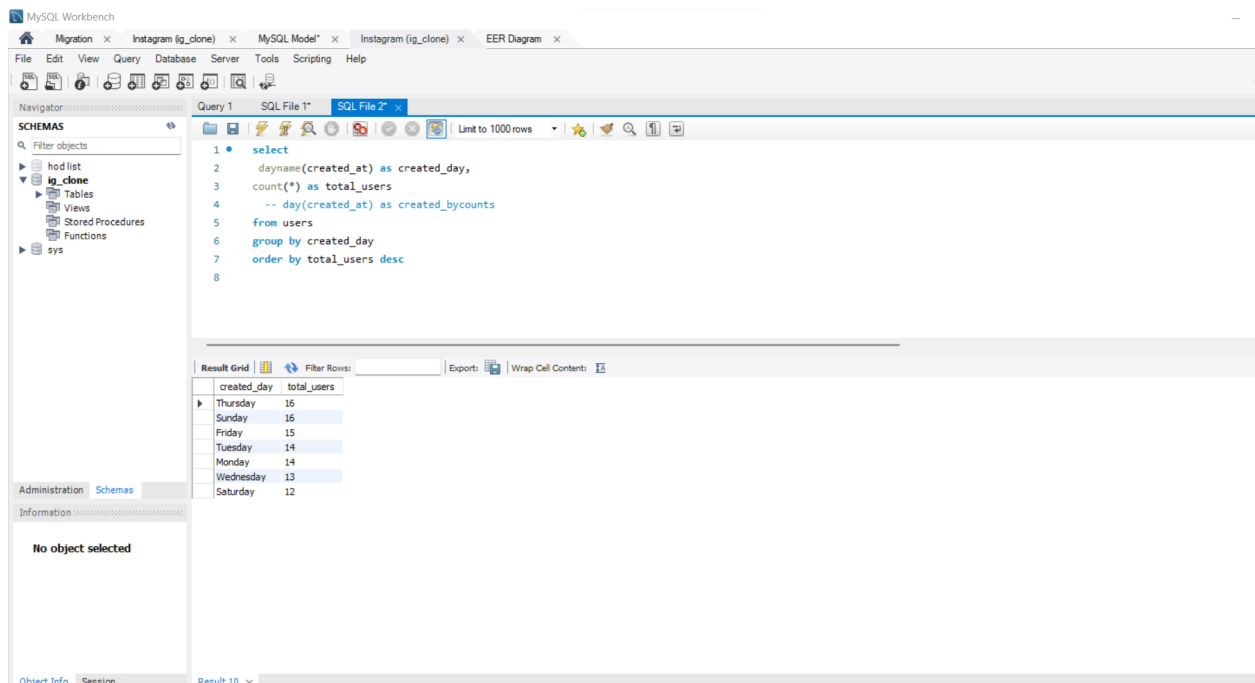
```

The 'Result Grid' shows the results of the query, limited to the top 5 hashtags. The results are as follows:

tag_name	tag_id	mosttag
smile	21	59
beach	20	42
party	17	39
fun	13	38
concert	18	24

5. Launch 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



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view showing 'hod list', 'ig_clone' (selected), 'Tables', 'Views', 'Stored Procedures', 'Functions', and 'sys'. The main editor window shows a SQL query in 'Query 1':

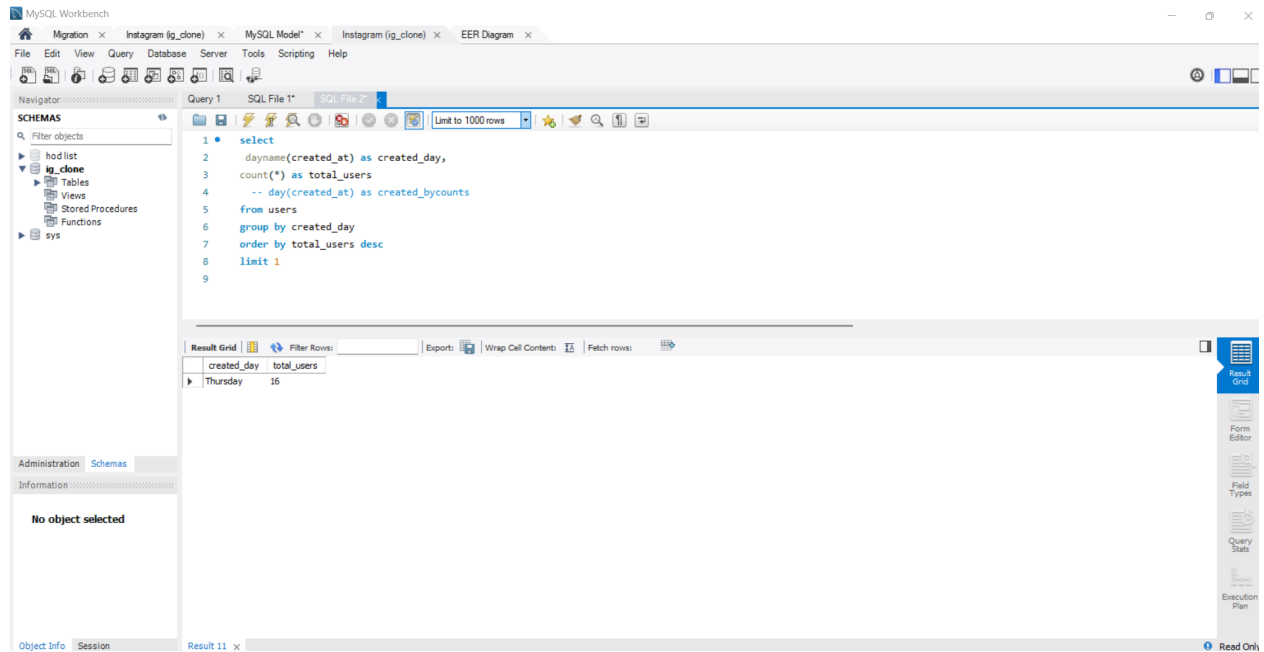
```
1 select
2   dayname(created_at) as created_day,
3   count(*) as total_users
4   -- day(created_at) as created_bycounts
5 from users
6 group by created_day
7 order by total_users desc
8
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The table has two columns: 'created_day' and 'total_users'. The results are as follows:

created_day	total_users
Thursday	16
Sunday	16
Friday	13
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

The bottom status bar shows 'Object Info' and 'Session' tabs, with 'Result 10' selected.

Insight :By putting a limit of 1 to get what day of the week most users registered on, since the values of Thursday and Sunday are equal based on the alphabetical Arrangement of the Words, Thursday gets the top rank. Therefore to finalize the day the Analyst team can further analyze which day to choose based on considering other factors.



The screenshot displays the MySQL Workbench interface. The 'Query 1' tab is active, showing the following SQL query:

```
1 select
2   dayname(created_at) as created_day,
3   count(*) as total_users
4   -- day(created_at) as created_bycounts
5 from users
6 group by created_day
7 order by total_users desc
8 limit 1
9
```

The 'Result Grid' at the bottom shows the output of the query:

created_day	total_users
Thursday	16

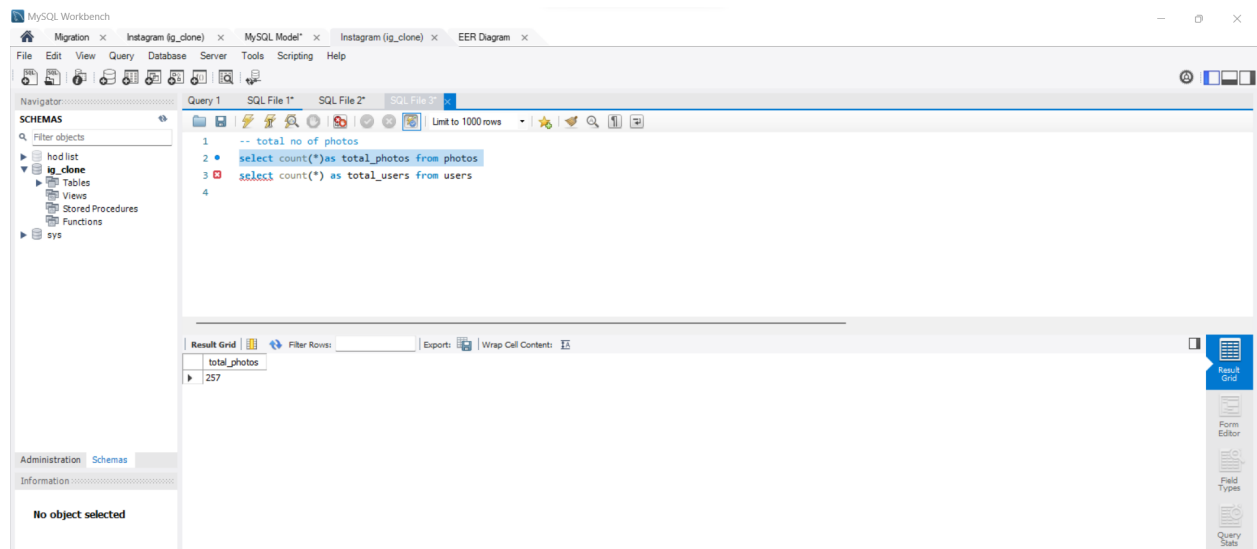
The interface also includes a 'Navigator' on the left showing the database schema, and a 'Read Only' button on the right.

INVESTOR METRICS

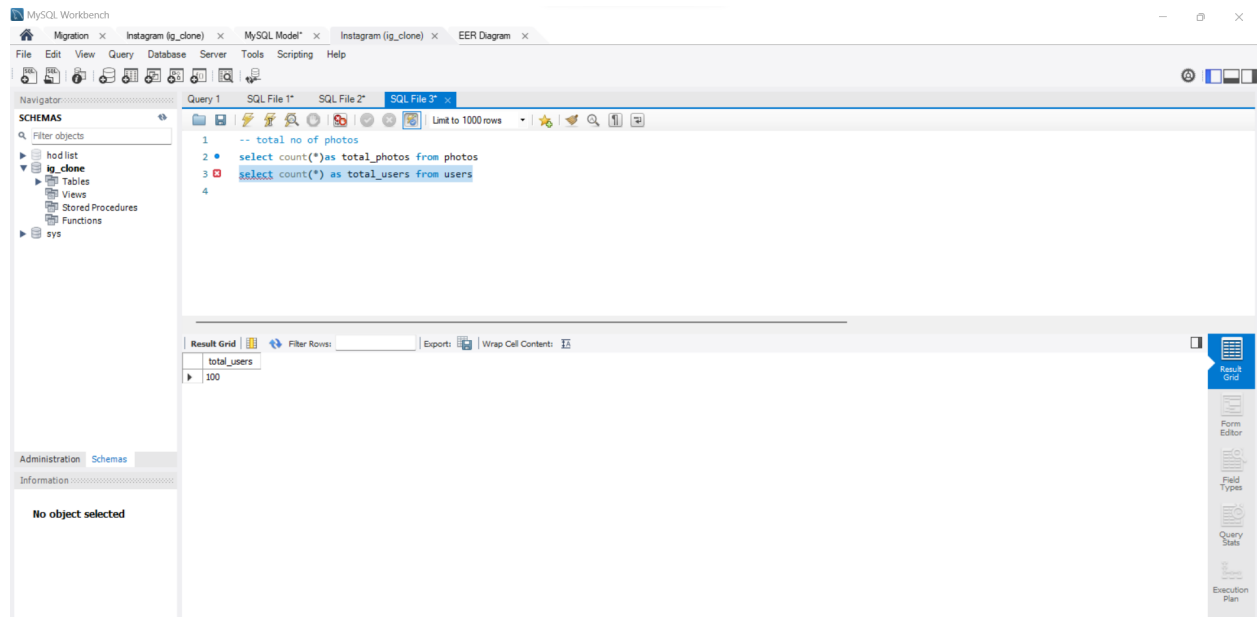
1.User Engagement: Are users still as active and post on Instagram or they are making fewer posts

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

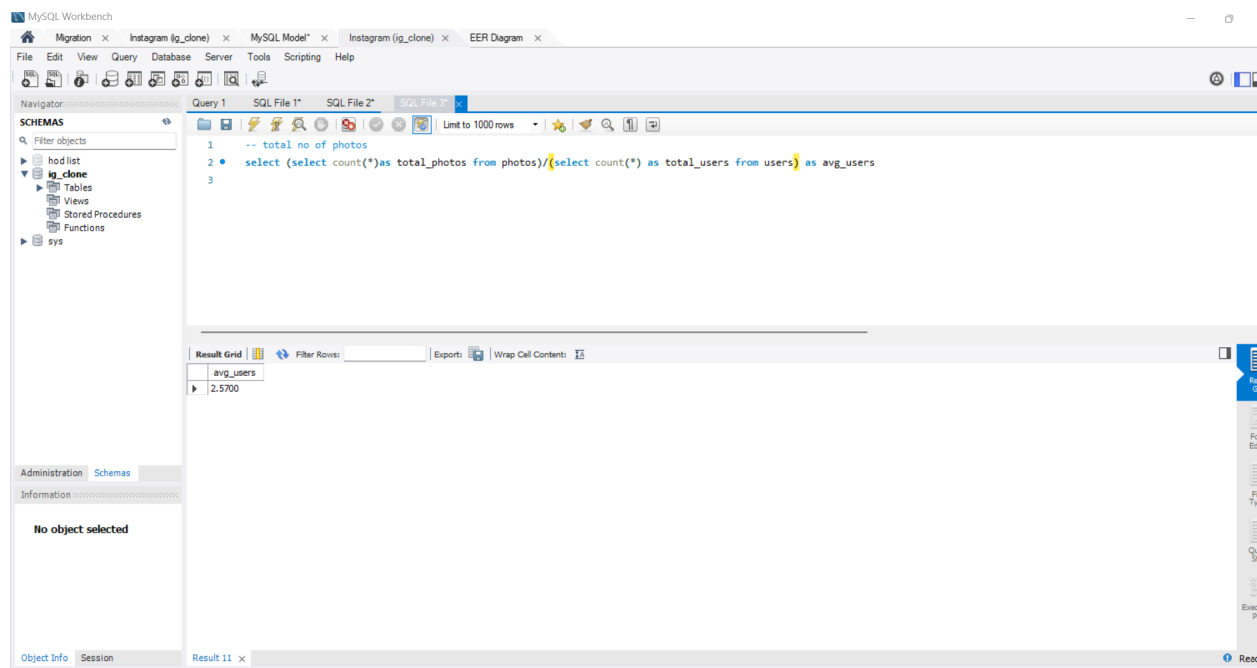
First total no of photos



Total numbers of users

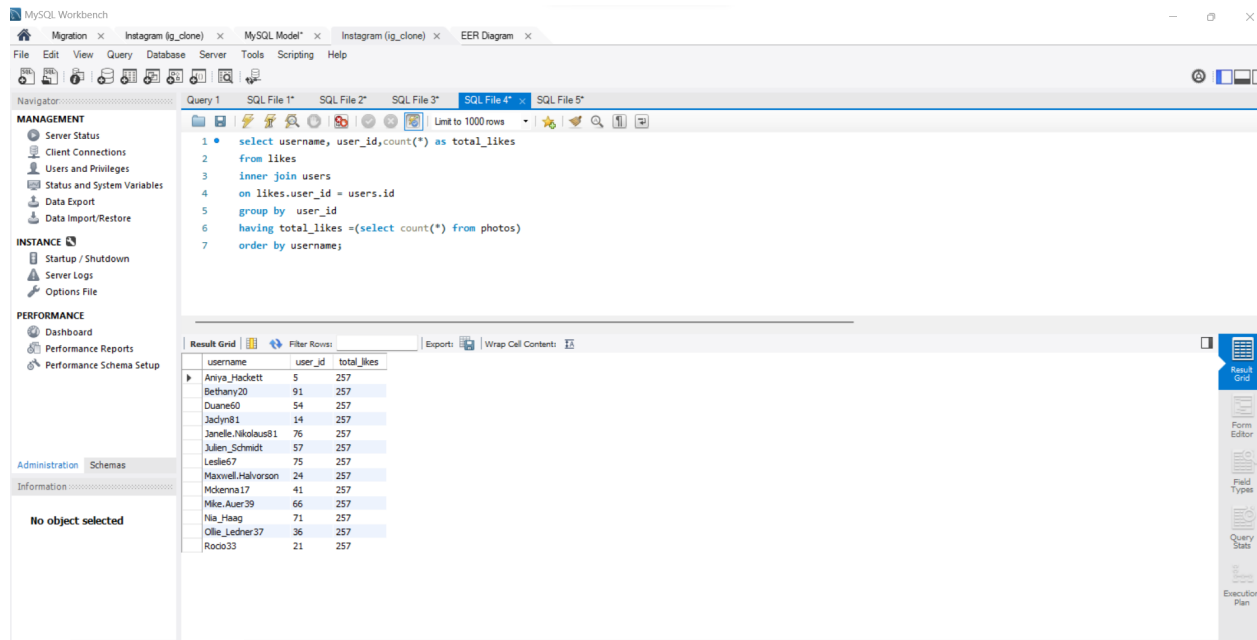


The average users post on Instagram



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).



Insight : In this code the “WHERE” clause cannot be used since where should be defined before group by and if we try to define it before group by the code would not be able to define total_likes therefore we use “HAVING”

Results

This Project helped me gain better understanding of
Joins
Sub-query
Count Clause

Having Clause

Mysql workbench

And a glimpse of working with a database how insightful a clean data can be, its usage in development product and marketing sectors of a company .