Instagram User Analytics project

Submitted by

Bishal kar

Ug student at NIT Agartala

Email: bishalkar875@gmail.com

Contact: -6009086878

Instagram worker analytics: -

1.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.

2.In this Instagram analytics project I have taken a database of Instagram and I have analysed It. There are some questions divided by two parts

1.Marketing
2. Investor Metrics

3. I solved this project by MQ SQL software.

Approach: -

1st of all I go through the given dataset and created the ig_clone database
By using create command.in this database I have created different tables
Like users, likes, photos, tags, comments etc. here I also use create command.
After creating database and tables I have insert the given values into the tables by using insert values command. after this I go through my task part.

Tech-stack used: -

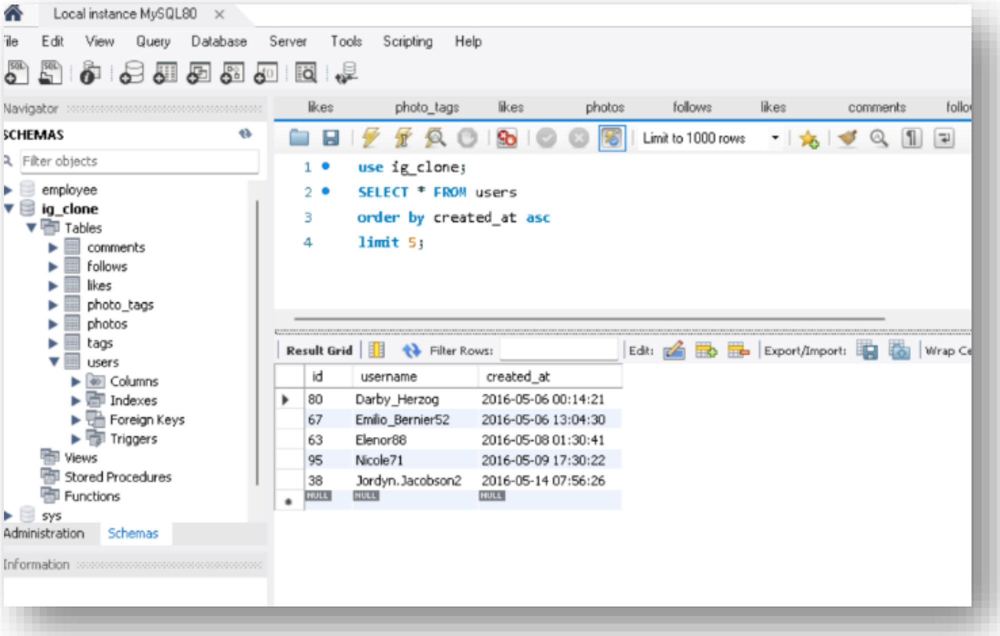
For this project I have used My SQL (8.0) software It is open source, reliable, compatible with all major hosting providers, cost-effective, and easy to manage. MySQL can run on very modest hardware and puts very little strain on system resources.

A.1) Rewarding Most Loyal Users:

So we have to find the 5 oldest users of Instagram from the given database. So for that I use users table. Selecting all from users I ordered it by the date of created_at (on which date account was created). So we have to select top 5 so we give limit 5.

Query: select * from users order by created_at limit 5;

Output is given below.



```
use ig_clone;
SELECT * FROM users
order by created_at asc
limit 5;
```

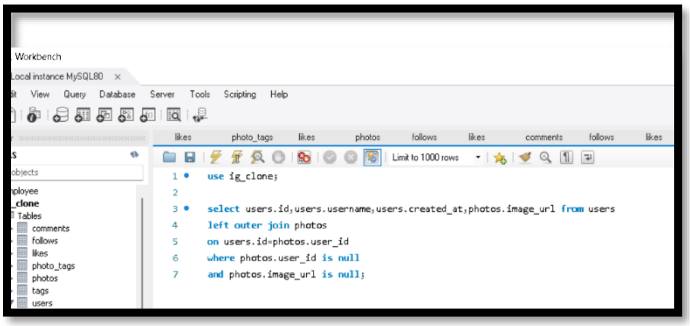
id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

2) Remind Inactive Users to Start Posting: -

So we have to Find the users who have never posted a single photo on Instagram.

For this we are using left outer join between users and photos tables and here we see users.id=photos.user_id. now we find the peoples who never posted a single pic by using where photos.user_id is null and photos.image_url is null.

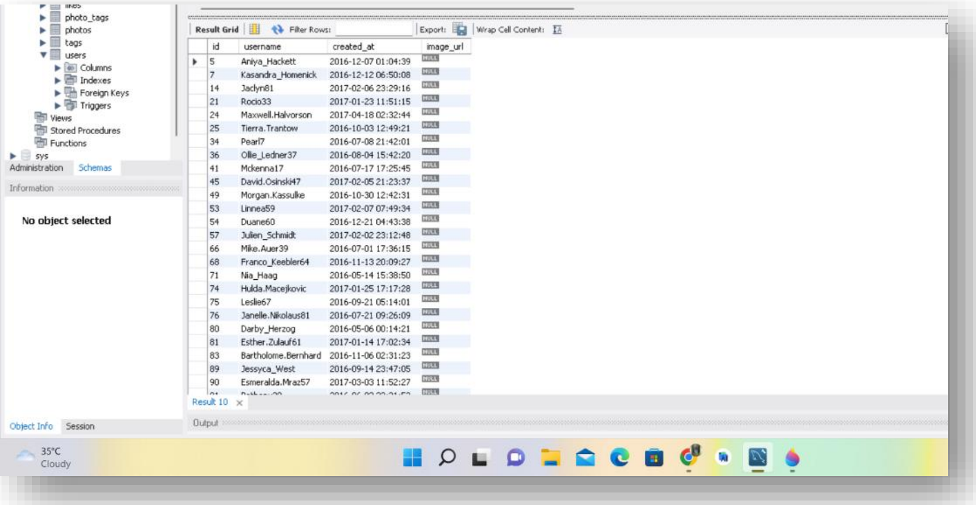
```
Query:selectusers.id,users.uysername,users.created_at,photos.image
_url from users
left outer join photos
on users.id = photos.user_id
Where photos.user.id is null
 and photos.image_url is null;
```



```sql
use ig_clone;

select users.id,users.username,users.created_at,photos.image_url from users
left outer join photos
on users.id=photos.user_id
where photos.user_id is null
and photos.image_url is null;
```

Output is given below.



id	username	created_at	image_url
5	Aniya_Hackett	2016-12-07 01:04:39	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL
14	Jaclyn81	2017-02-06 23:29:16	NULL
21	Rocio33	2017-01-23 11:51:15	NULL
24	Maxwell.Halvorson	2017-04-18 02:32:44	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL
34	Pearl7	2016-07-08 21:42:01	NULL
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL
41	Mckenna17	2016-07-17 17:25:45	NULL
45	David.Osinski47	2017-02-05 21:23:37	NULL
49	Morgan.Kassulke	2016-10-30 12:42:31	NULL
53	Linnea59	2017-02-07 07:49:34	NULL
54	Duane60	2016-12-21 04:43:38	NULL
57	Julien_Schmidt	2017-02-02 23:12:48	NULL
66	Mike.Auer39	2016-07-01 17:36:15	NULL
68	Franco_Keebler64	2016-11-13 20:09:27	NULL
71	Nia_Haag	2016-05-14 15:38:50	NULL
74	Hulda.Macejkovic	2017-01-25 17:17:28	NULL
75	Leslie67	2016-09-21 05:14:01	NULL
76	Janelle.Nikolaus81	2016-07-21 09:26:09	NULL
80	Darby_Herzog	2016-05-06 00:14:21	NULL
81	Esther.Zulauf61	2017-01-14 17:02:34	NULL
83	Bartholome.Bernhard	2016-11-06 02:31:23	NULL
89	Jessyca_West	2016-09-14 23:47:05	NULL
90	Esmeralda.Mraz57	2017-03-03 11:52:27	NULL

3) Declaring Contest Winner: -

We have to find the user who gets the most likes on a single photo. for this we using join operation between likes and photos.selecting user_id,id from photos the new name of id is photos_id and count the total photos and coloumn name is most_liked here we see likes.photos_id=photos.id.after using join operation I Use group by and oder by operation

Query: select photos.user_id,photos.id as photos_id,count(*) as most_liked

From likes

Join photos

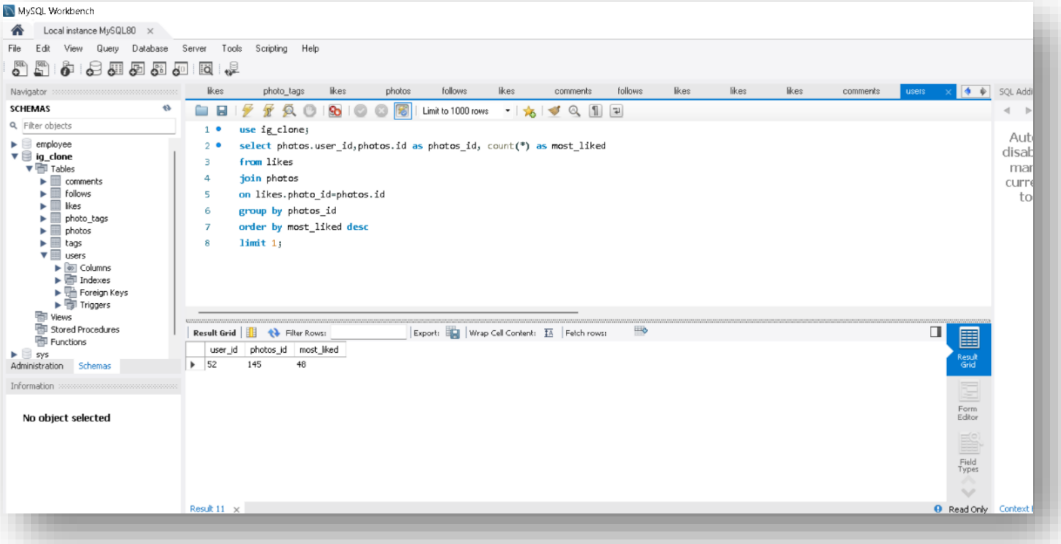
On likes.photo_id=photos.id

Group by photos_id

Order by most_liked desc

Limit 1;

Output is given below.



```
use ig_clone;
select photos.user_id,photos.id as photos_id, count(*) as most_liked
from likes
join photos
on likes.photo_id=photos.id
group by photos_id
order by most_liked desc
limit 1;
```

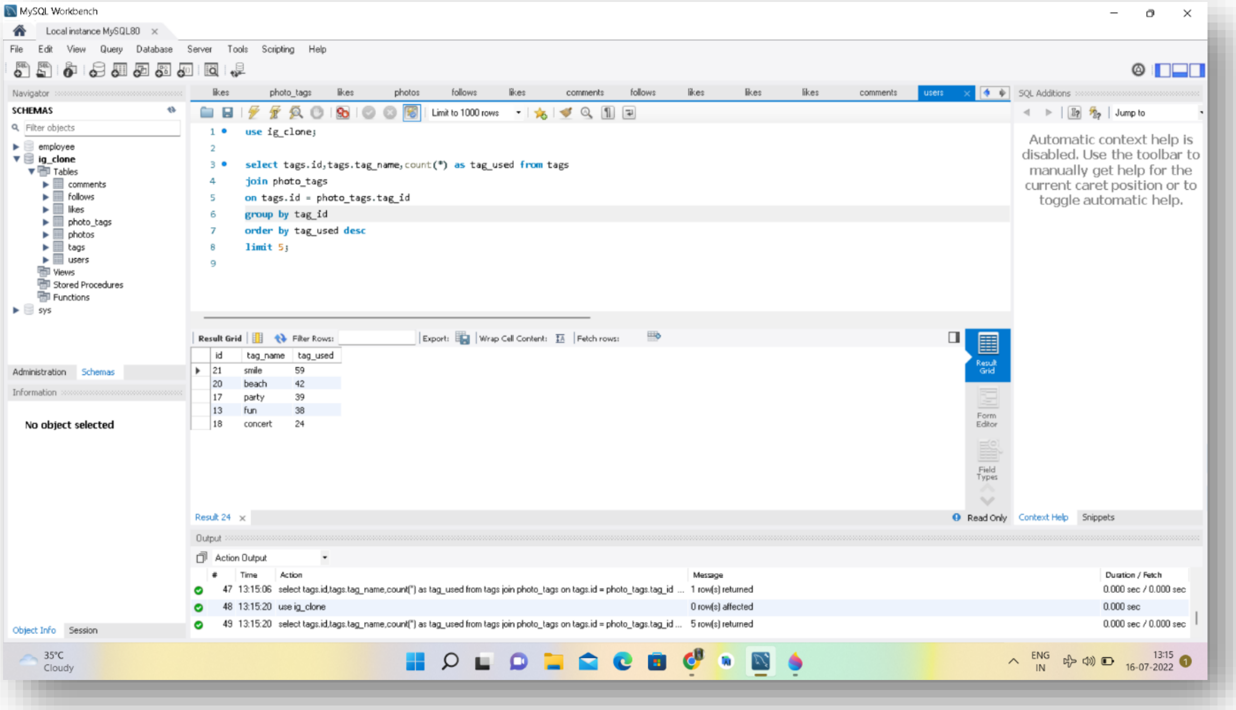
user_id	photos_id	most_liked
52	145	48

4) Hashtag Researching:

A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform. So I have to Identify and suggest the top 5 most commonly used hashtags on the platform. So I used photo_tags table. 1 st of all I select tags.tags_name and named as tag_used. Then join tags on photo_tags.tag_id = tag.id. Then I group by tags_id. And order tag_used in DESC. And as I have to find top most commonly used hash tags so I put limit 5.

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

Output is given below.



```
use ig_clone;

select tags.id,tags.tag_name,count(*) as tag_used from tags
join photo_tags
on tags.id = photo_tags.tag_id
group by tag_id
order by tag_used desc
limit 5;
```

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

```
#   Time      Action                                                                                    Message            Duration / Fetch
47  13:15:06  select tags.id,tags.tag_name,count(*) as tag_used from tags join photo_tags on tags.id = photo_tags.tag_id ...  1 row(s) returned  0.000 sec / 0.000 sec
48  13:15:20  use ig_clone                                                                              0 row(s) affected  0.000 sec
49  13:15:20  select tags.id,tags.tag_name,count(*) as tag_used from tags join photo_tags on tags.id = photo_tags.tag_id ...  5 row(s) returned  0.000 sec / 0.000 sec
```

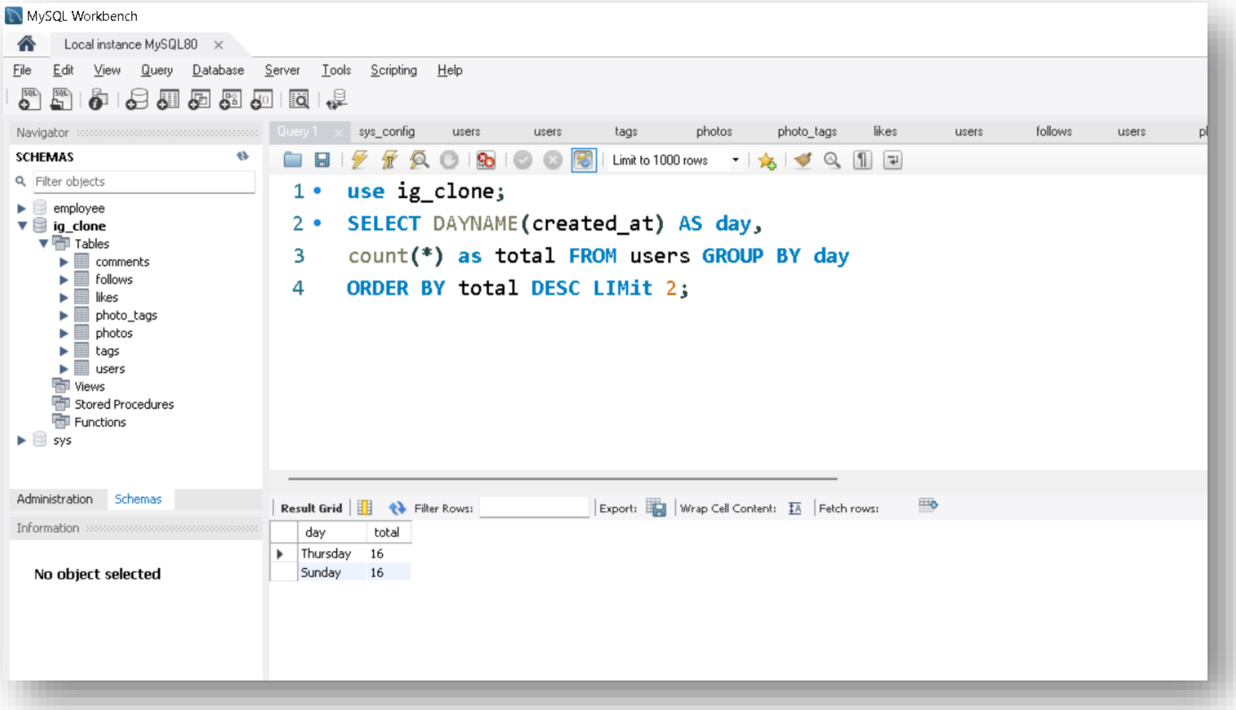
5) Launch AD Campaign:

The team wants to know, which day would be the best day to launch ADs. What day of the week do most users register on? Provide insights on when to schedule and campaign. So for that I select dayname Ie created at as day then count as total from users. Then group by day then order by total. Then use DESC and give limit 2.

Query: SELECT DAYNAME(created_at) AS day, count(*) as total FROM users GROUP BY day

ORDER BY total DESC LIMIT 2;

Output is given below.



```
use ig_clone;
SELECT DAYNAME(created_at) AS day,
count(*) as total FROM users GROUP BY day
ORDER BY total DESC LIMIt 2;
```

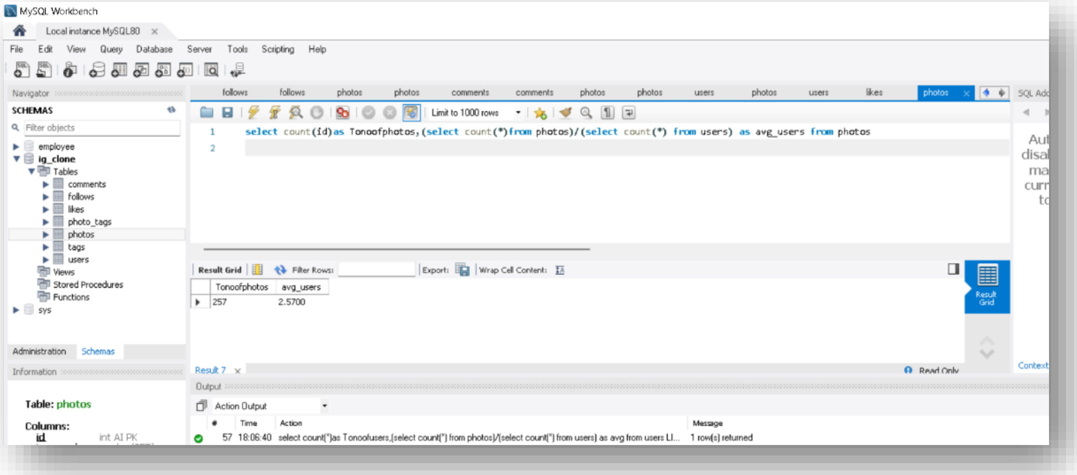
day	total
Thursday	16
Sunday	16

B.1) User Engagement:

Are users still as active and post on Instagram or they are making fewer posts I have to Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users. So 1st of all I select all from photos then divided it by selecting all from users. Then select the whole and name as avg.

Query: SELECT count(id) as tonoofphotos,
 (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS avg
From photos;

Output is given below.



```
select count(id)as Tonoofphotos,(select count(*)from photos)/(select count(*) from users) as avg_users from photos
```

```
Tonoofphotos  avg_users
257           2.5700
```

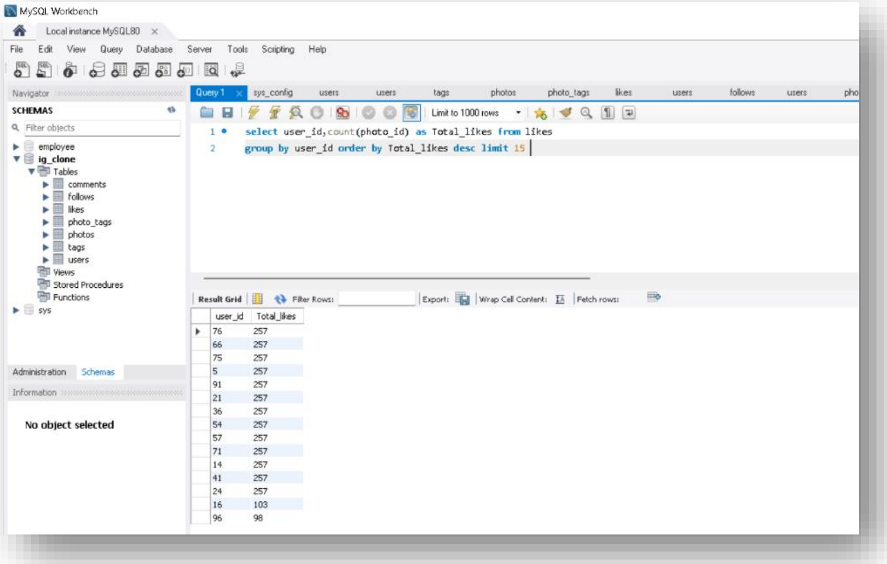
```
#   Time      Action                                                                                          Message
57  18:06:40  select count(*)as Tonoofusers,(select count(*) from photos)/(select count(*) from users) as avg from users LI...  1 row(s) returned
```

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). So 1st of all we use likes table we select user_id and count photo id and named as total likes then after some query operations we found that 13 users have liked all the posts. Like count 257.

Query: select user_id,count(photo_id) as Total_Likes from likes where user_id in (select user_id from likes) group by user_id order by count(photo_id) DESC limit 1

Output is given below.



Limit to 1000 rows

```sql
select user_id,count(photo_id) as Total_likes from likes
group by user_id order by Total_likes desc limit 15
```

user_id	Total_likes
76	257
66	257
75	257
5	257
91	257
21	257
36	257
54	257
57	257
71	257
14	257
41	257
24	257
16	103
96	98

Thank you…