

# INSTAGRAM USER ANALYTICS

---

## **PROJECT DESCRIPTION**

Here we are trying to build a system through which we will be able to build better business opportunities and better client interaction.

Through this user analytics we will measure certain parameters which in turn will help us grow as a company and also most certainly help in understanding our users. By this way we would also be able to recommend best of our services to our users. These users analytics are the foundation pillar of the success of any organisation. All the activities performed by users, like and dislikes of users every minute detail is paid attention and hence helps in creating a best possible service to the users.

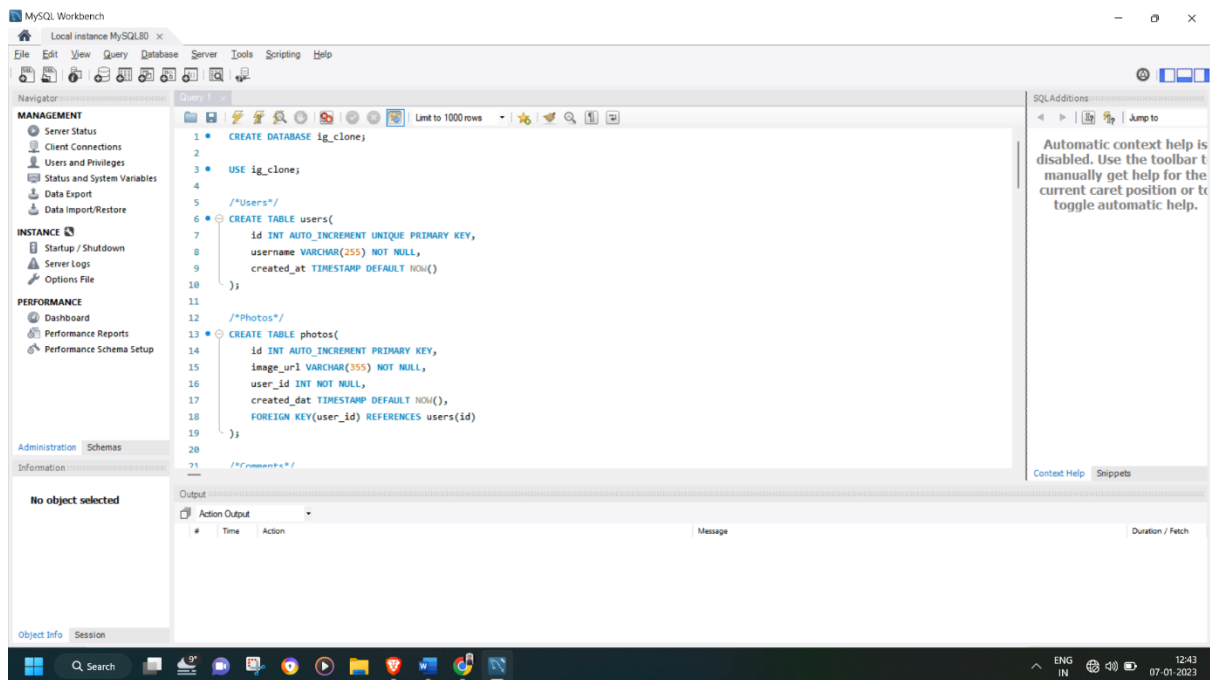
In this project we are going to work out with a dataset of instagram users which contains various parameters such as registering on the app, liking a photo, using hashtags etc. With all these provided data we would try to derive the outcomes desired from us.

## **APPROACH**

I have tried to understand the dataset before trying to execute any of the query. I related each given data with what exactly I require to derive e.g. If I have been given the dataset of user registration and asked to find out the oldest user then I searched for the user having oldest date of registration. So over in all my approach was quite simple I just kept on connecting the dots to build these query list.

# TECH-STACK

I had used MySQL Workbench 8.0 CE available in MySQL.com. It is a unified visual tool for database architects, developers and DBA's.



This is a screenshot of the interface of MySQL Workbench. The reason for using it is that it has very user friendly interface and it is also hassle free with all the provided services such as creating a database, administering it, modifying it etc. I have particularly used it to create a database named ig\_clone and then I have used multiple queries to derive the outcome I required out of the given dataset.

## **INSIGHTS**

I have gathered a great deal of knowledge from this assignment. I got a real exposure of how exactly an organisation works to get close to its client and generate the best revenue out of the services or goods it offers. The ways in which here it is mentioned about the requirement from the given dataset it give you the opportunity to map the data like I had to find the most used hashtags so first used the count function to get that which hashtags is used how many times and then arrange it in descending order using order by function to help get the top 5 hashtags which is most used. Certainly there were other function such as join function, max, limit etc. These functions help in deriving the sorted data out of such a huge provided data.

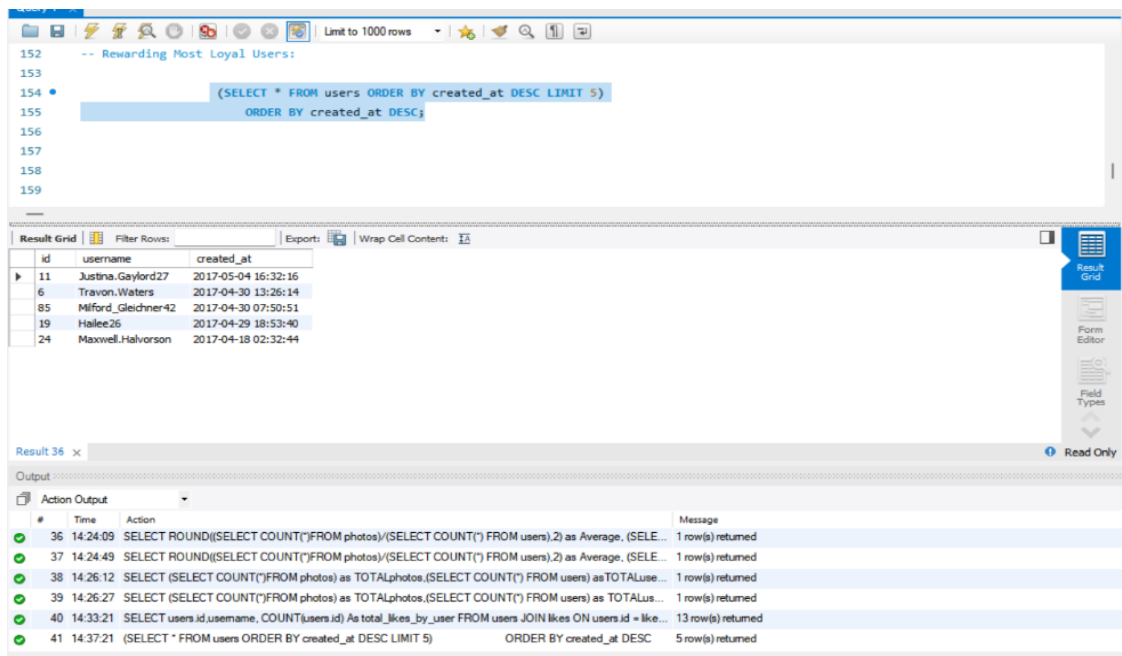
In short these queries help us in getting the exact same amount of data or information we require out of the whole dataset.

# RESULT

## A.) MARKETING

### 1.) Rewarding Most Loyal Users:

(SELECT \* FROM users ORDER BY created\_at DESC LIMIT 5)  
ORDER BY created\_at DESC;



The screenshot shows a database management interface. At the top, a SQL query is entered in a text area: `-- Rewarding Most Loyal Users: (SELECT * FROM users ORDER BY created_at DESC LIMIT 5) ORDER BY created_at DESC;`. Below the query, a 'Result Grid' displays the results of the query. The grid has three columns: 'id', 'username', and 'created\_at'. The results are as follows:

id	username	created_at
11	Justina.Gaylord27	2017-05-04 16:32:16
6	Travon.Waters	2017-04-30 13:26:14
85	Miford_Gleichner42	2017-04-30 07:50:51
19	Hallee26	2017-04-29 18:53:40
24	Maxwell.Halvorsen	2017-04-18 02:32:44

Below the result grid, an 'Output' pane shows a log of database actions. The last action is: `(SELECT * FROM users ORDER BY created_at DESC LIMIT 5) ORDER BY created_at DESC`, which returned 5 rows.

### 2.) Remind Inactive Users to Start Posting:

SELECT \*  
FROM users  
LEFT JOIN photos ON users.id = photos.user\_id  
WHERE photos.id IS NULL;

152 -- Remind Inactive Users to Start Posting:  
 153  
 154 • `select * from users left join photos`  
 155 `on photos.user_id=users.id`  
 156 `where created_dat is Null;`  
 157  
 158  
 159  
 160

id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jadyn81	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 17:25:45	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

Result 37 x

Output

#	Time	Action	Message
37	14:24:49	SELECT ROUND((SELECT COUNT(*) FROM photos)/(SELECT COUNT(*) FROM users),2) as Average, (SELE...	1 row(s) returned
38	14:26:12	SELECT (SELECT COUNT(*) FROM photos) as TOTAL photos (SELECT COUNT(*) FROM users) as TOTAL users	1 row(s) returned

### 3.) Declaring Contest Winner:

```
SELECT username,photos.id,photos.image_url,
COUNT(*) AS total
FROM photos
INNER JOIN likes
ON likes.photo_id = photos.id
INNER JOIN users
ON photos.user_id = users.id
GROUP BY photos.id
ORDER BY total DESC
LIMIT 1;
```

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
-- Declaring Contest Winner:  
  
SELECT username, photos.id, photos.image_url, COUNT(*) AS total  
FROM photos  
INNER JOIN likes  
ON likes.photo_id = photos.id  
INNER JOIN users  
ON photos.user_id = users.id  
GROUP BY photos.id  
ORDER BY total DESC  
LIMIT 1;
```

The results pane shows a single row of data:

username	id	image_url	total
Zack_Kemmer93	145	https://jarret.name	48

The interface includes a toolbar at the top with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button on the right. The bottom status bar indicates 'Result 39' and 'Read On'.

#### 4.) Hashtag Researching:

```
SELECT tag_name, COUNT(tag_name) AS total  
FROM tags  
JOIN photo_tags ON tags.id = photo_tags.tag_id  
GROUP BY tags.id  
ORDER BY total DESC limit 5;
```

The screenshot shows a database query editor interface. The main area contains a SQL query for hashtag research. Below the query, the 'Result Grid' shows the top 5 results. The bottom section displays the 'Action Output' with a log of executed queries and their results.

```

156
157
158 -- Hashtag Researching:
159
160 • SELECT tag_name, COUNT(tag_name) AS total
161 FROM tags
162 JOIN photo_tags ON tags.id = photo_tags.tag_id
163 GROUP BY tags.id
164 ORDER BY total DESC limit 5;
165
166
167
168
169
170
171

```

tag_name	total
smile	59
beach	42
party	39
fun	38
concert	24

Result 42 x Read Only

Output

Action Output

#	Time	Action	Message
46	14:59:47	SELECT * FROM users LEFT JOIN photos ON users.id = photos.user_id WHERE photos.id IS NULL LIMIT 0, ...	26 row(s) returned
47	15:05:30	SELECT tag_name, COUNT(tag_name) AS total FROM tags JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tag_name ORDER BY total DESC limit 5;	5 row(s) returned

## 5.) Launch AD Campaign:

SELECT DAYNAME(created\_at) AS day,

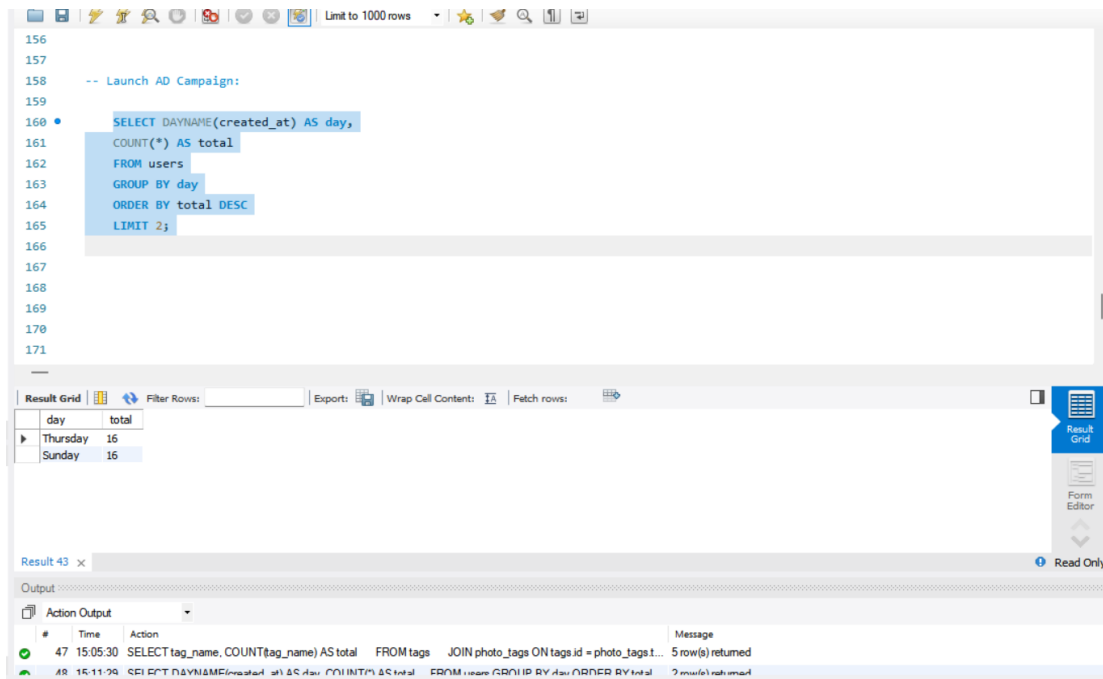
COUNT(\*) AS total

FROM users

GROUP BY day

ORDER BY total DESC

LIMIT 2;

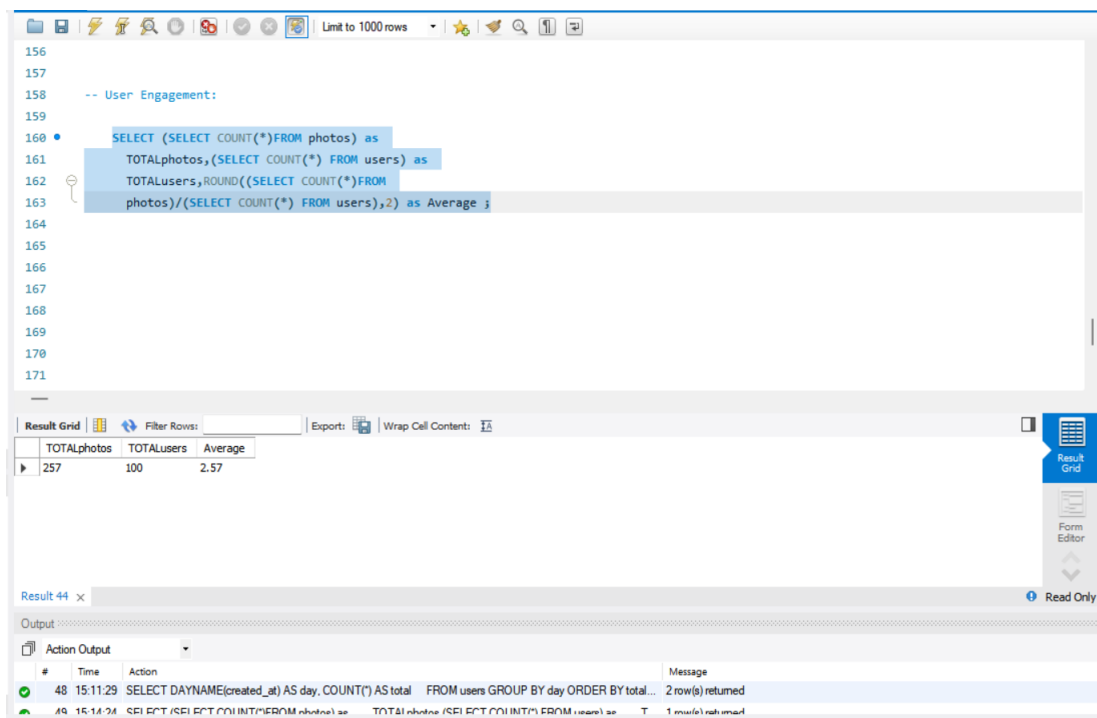


## **B.) INVESTOR METRICS**

### **1.) User Engagement:**

SELECT (SELECT COUNT(\*)FROM photos) as  
TOTALphotos,(SELECT COUNT(\*) FROM users) as  
TOTALUsers,ROUND((SELECT COUNT(\*)FROM  
photos)/(SELECT COUNT(\*) FROM users),2) as Average ;



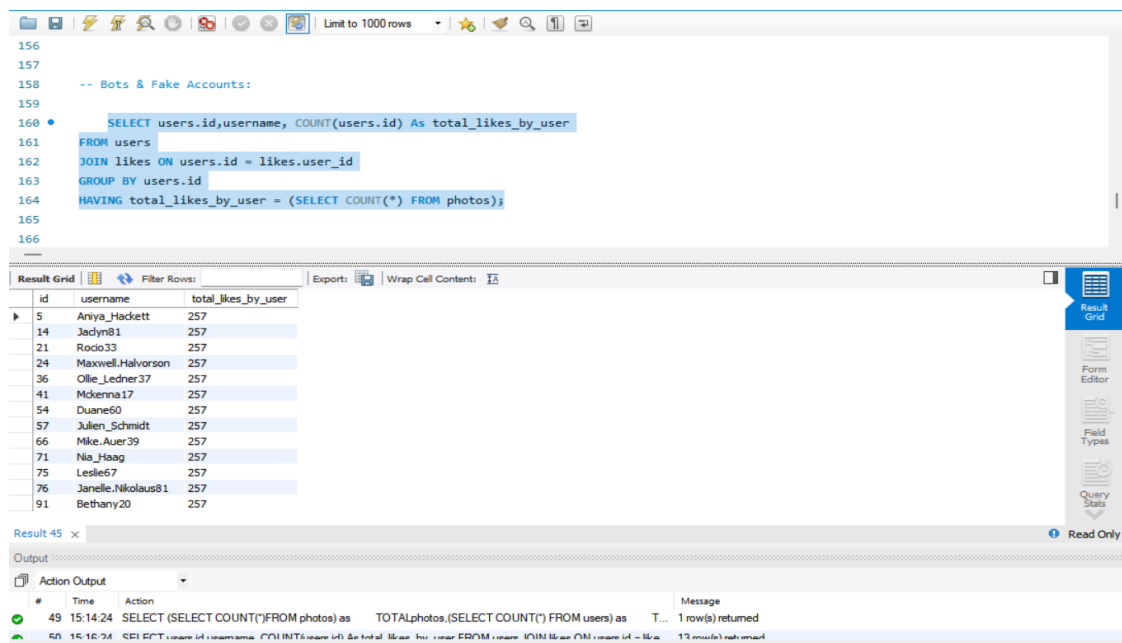


## 2.) Bots & Fake Accounts:

```

SELECT users.id, username, COUNT(users.id) As
total_likes_by_user
FROM users
JOIN likes ON users.id = likes.user_id
GROUP BY users.id
HAVING total_likes_by_user = (SELECT COUNT(*)
FROM photos);

```



I have successfully been able to segregate the dataset with what I required using exact query to differentiate. I have rewarded the 5 oldest user using Instagram, I have found the list of inactive users of Instagram and pinned them to use it through a personalized email, I have successfully found the winner of most liked photo contest, I have found the top 5 trending hashtags, I have found the best possible day to upload an ad for the marketing team. Moreover for the stats I have also found whether the user are active or not finding the average post, I have also successfully found all the fake accounts to maintain Instagram's credibility.

It gave me a great amount of pleasure to complete the project I learned all the functions used in queries with a live implication it get etched in my memory easily, I also learnt about aggregate functions.



**Thank  
You!!!**

www.dreamstime.com/stock-illustration-image

Submitted by :- **Bishwomkar Panigrahi**