Project no 2

Instagram user analytics

Using SQL and MySQL Workbench

Name: SUNIL KUMAR NALLABOTHULA

Mail: johnmarkjmk3092020@gmial.com

1. Project description:

As far as we believe that the Users or the customers or the consumers and satisfying them with our product or service, we have to first understand them and give some loyalty to them as they are giving us.
This project is mainly focus on Understanding the user and their behavior towards our app and service etc.
It includes finding inactive of them and giving some loyalty prizes, encouraging them to participate in some activities etc.
Depends up on we'll come to a conclusion where ads campaigns need to be launched and detecting bots and fake accounts.

2. Approach:

In this project we are going to use all the necessary data of users. And has to load into SQL Workbench and perform some queries to get some useful insights and patterns to answer following tasks and questions.

Task A:

1. Loyal user reward
2. Inactive user engagement
3. Contest winner declaration
4. Hashing Research
5. Ad campaign

Task B:

1. User Engagement
2. Finding Bots & Fake account

Now let's see the solutions for all the above questions and problems;

Task A;

Question 1; **Loyal user reward**

Identifying the Five oldest users on Instagram and assigning to them with Loyal user reward.

From the provided database we have known that these are the Five and Loyal Users.

id	User name
80	Darby_Herzog
67	Emilio_Bernier52
63	Elenor88
95	Nicole71
38	Jordyn.Jacobson2

- These are the SQL queries has run out to get results: -

Input;

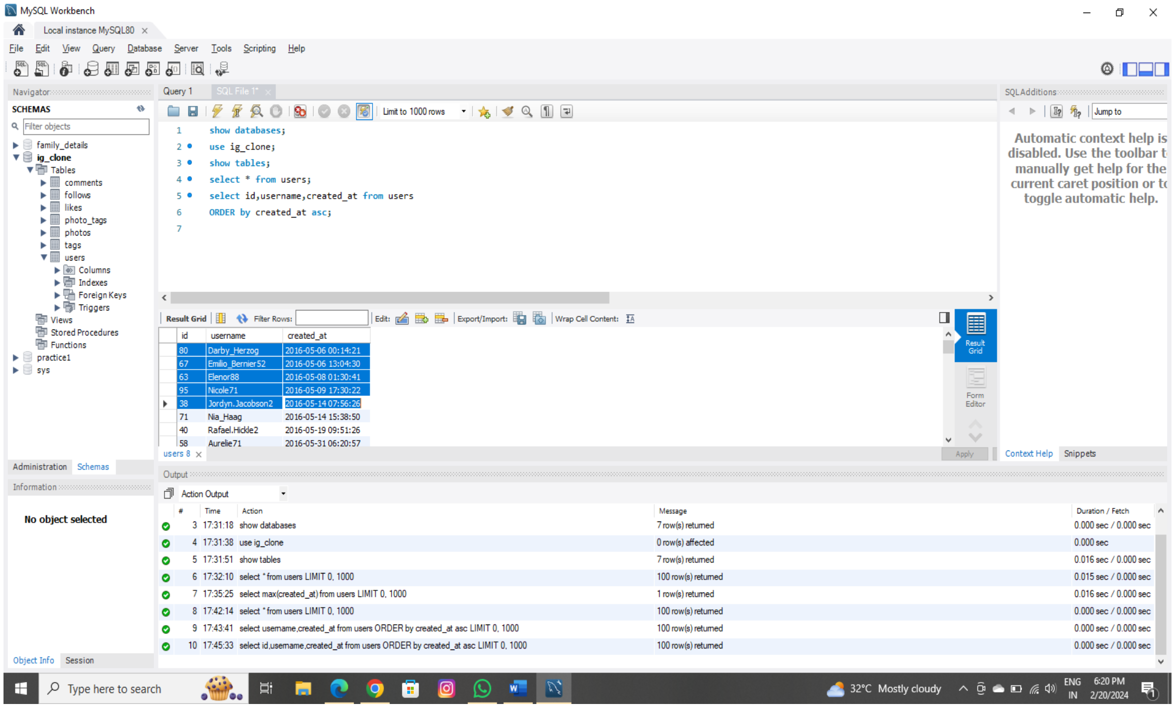
```
show databases;

use ig_clone;

show tables;
```

select * from users;

select id,username,created_at from users

ORDER by created_at asc

- This is the result screen shot



```
show databases;
use ig_clone;
show tables;
select * from users;
select id,username,created_at from users
ORDER by created_at asc;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.

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
71	Nia_Haag	2016-05-14 15:38:50
40	Rafael.Hickle2	2016-05-19 09:51:26
58	Aurelie71	2016-05-31 06:20:57

#	Time	Action	Message	Duration / Fetch
3	17:31:18	show databases	7 row(s) returned	0.000 sec / 0.000 sec
4	17:31:38	use ig_clone	0 row(s) affected	0.000 sec
5	17:31:51	show tables	7 row(s) returned	0.016 sec / 0.000 sec
6	17:32:10	select * from users LIMIT 0, 1000	100 row(s) returned	0.015 sec / 0.000 sec
7	17:35:25	select max(created_at) from users LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
8	17:42:14	select * from users LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
9	17:43:41	select username,created_at from users ORDER by created_at asc LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
10	17:45:33	select id,username,created_at from users ORDER by created_at asc LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

Question 2; Inactive User Engagement

Identifying the users who did not post any single one since from started his account.

From the provided database we have known that these are the Inactive Users who did not post anything yet.

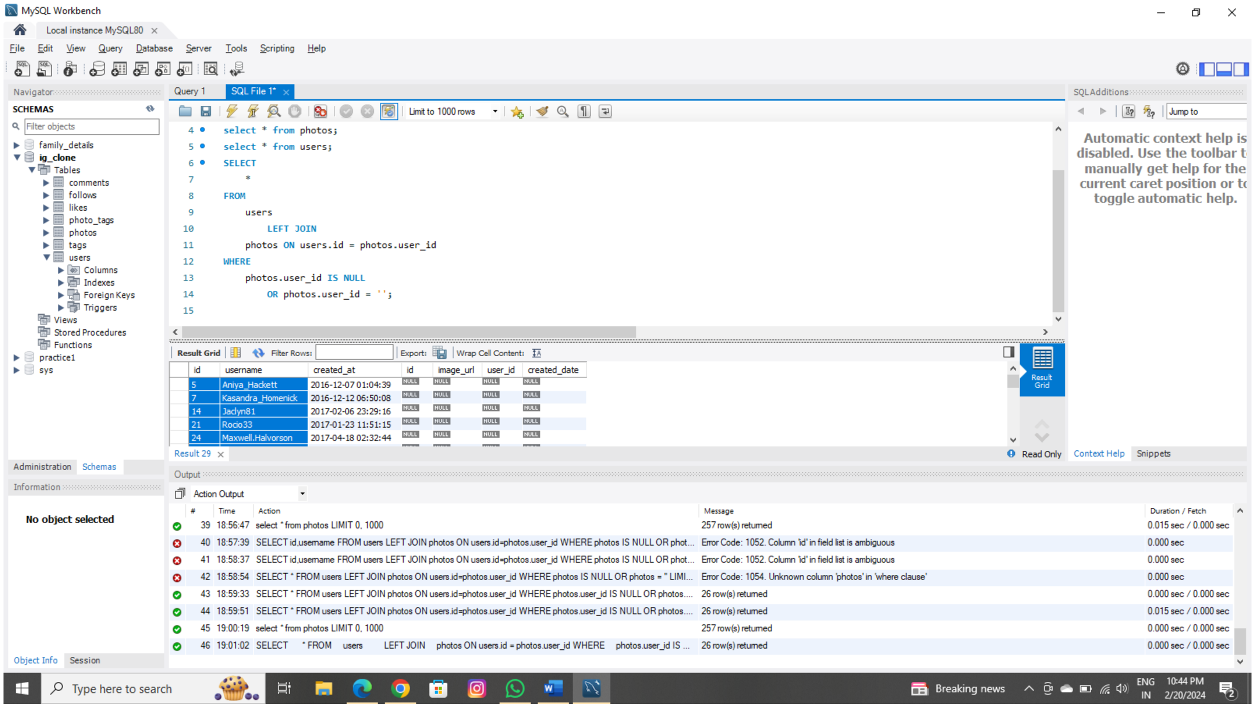
Use_id	User_name
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

❖ These are the SQL queries has run out to get results: -

```
SELECT
    *
```

```
FROM
    users
        LEFT JOIN
    photos ON users.id = photos.user_id
WHERE
    photos.user_id IS NULL
        OR photos.user_id = '';
```

These are the users those who are never active in the Instagram with their user id’s and user names with screen shot.



```
select * from photos;
select * from users;
SELECT
    *
FROM
    users
        LEFT JOIN
    photos ON users.id = photos.user_id
WHERE
    photos.user_id IS NULL
        OR photos.user_id = '';
```

id	username	created_at	id	image_url	user_id	created_date
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	Jaclyn81	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

```
#   Time      Action                                                                                      Message                                                      Duration / Fetch
39  18:56:47  select * from photos LIMIT 0, 1000                                                          257 row(s) returned                                          0.015 sec / 0.000 sec
40  18:57:39  SELECT id,username FROM users LEFT JOIN photos ON users.id=photos.user_id WHERE photos IS NULL OR phot...   Error Code: 1052. Column 'id' in field list is ambiguous   0.000 sec
41  18:58:37  SELECT id,username FROM users LEFT JOIN photos ON users.id=photos.user_id WHERE photos IS NULL OR phot...   Error Code: 1052. Column 'id' in field list is ambiguous   0.000 sec
42  18:58:54  SELECT * FROM users LEFT JOIN photos ON users.id=photos.user_id WHERE photos IS NULL OR photos = '' LIMI...   Error Code: 1054. Unknown column 'photos' in 'where clause'   0.000 sec
43  18:59:33  SELECT * FROM users LEFT JOIN photos ON users.id=photos.user_id WHERE photos.user_id IS NULL OR photos....   26 row(s) returned   0.000 sec / 0.000 sec
44  18:59:51  SELECT * FROM users LEFT JOIN photos ON users.id=photos.user_id WHERE photos.user_id IS NULL OR photos....   26 row(s) returned   0.015 sec / 0.000 sec
45  19:00:19  select * from photos LIMIT 0, 1000                                                          257 row(s) returned                                          0.000 sec / 0.000 sec
46  19:01:02  SELECT    * FROM    users      LEFT JOIN    photos ON users.id = photos.user_id WHERE    photos.user_id IS ...   26 row(s) returned   0.000 sec / 0.000 sec
```

Question 3; contest winner Declaration

Determining the winner of the contest and providing their details;

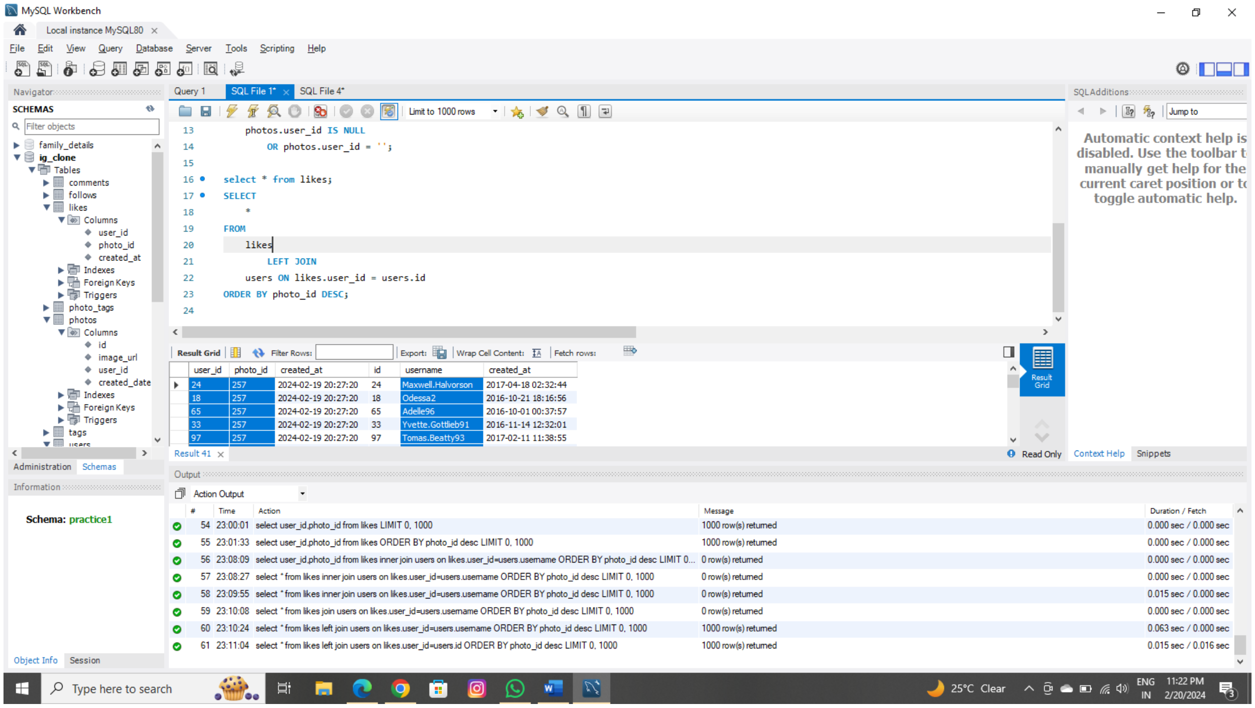
From the provided database we have obtained that these are the Winner of the contest and their details.

User_id	No_of_likes	User_name
24	257	Maxwell.Halvorson
18	257	Odessa2
65	257	Adelle96
33	257	Yvette.Gottlieb91
97	257	Tomas.Beatty93
41	257	Mckenna17
39	257	Kelsi26
3	257	Harley_Lind18
12	257	Dereck65
91	257	Bethany20
73	257	Jaylan.Lakin
66	257	Mike.Auer39
61	257	Jayson65
75	257	Leslie67
14	257	Jaclyn81
21	257	Rocio33
15	257	Billy52
54	257	Duane60
57	257	Julien_Schmidt
70	257	Erick5
60	257	Sam52
43	257	Janet.Armstrong
31	257	Aiyana_Hoeger
5	257	Aniya_Hackett
71	257	Nia_Haag
35	257	Lennie_Hartmann40
95	257	Nicole71
47	257	Harrison.Beatty50
82	257	Aracely.Johnston98
48	257	Granville_Kutch
76	257	Janelle.Nikolaus81
10	257	Presley_McClure

26	257	Josianne.Friesen
36	257	Ollie_Ledner37
92	257	Frederik_Rice
20	257	Delpha.Kihn

- ❖ above information is all about the users and their names who got highest likes and eligible for winner of the contest.

- ❖ This is the screen shot of the quires has returned to get them



```
    photos.user_id IS NULL
        OR photos.user_id = '';

select * from likes;
SELECT
    *
FROM
    likes
        LEFT JOIN
    users ON likes.user_id = users.id
ORDER BY photo_id DESC;
```

user_id	photo_id	created_at	id	username	created_at
24	257	2024-02-19 20:27:20	24	Maxwell.Halvorson	2017-04-18 02:32:44
18	257	2024-02-19 20:27:20	18	Odessa2	2016-10-21 18:16:56
65	257	2024-02-19 20:27:20	65	Adelle96	2016-10-01 00:37:57
33	257	2024-02-19 20:27:20	33	Yvette.Gottlieb91	2016-11-14 12:32:01
97	257	2024-02-19 20:27:20	97	Tomas.Beatty93	2017-02-11 11:38:55

```
#   Time      Action                                                                                                        Message                 Duration / Fetch
54  23:00:01  select user_id,photo_id from likes LIMIT 0, 1000                                                              1000 row(s) returned    0.000 sec / 0.000 sec
55  23:01:33  select user_id,photo_id from likes ORDER BY photo_id desc LIMIT 0, 1000                                        1000 row(s) returned    0.000 sec / 0.000 sec
56  23:08:09  select user_id,photo_id from likes inner join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0...  0 row(s) returned    0.000 sec / 0.000 sec
57  23:08:27  select * from likes inner join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000        0 row(s) returned       0.000 sec / 0.000 sec
58  23:09:55  select * from likes inner join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000        0 row(s) returned       0.015 sec / 0.000 sec
59  23:10:08  select * from likes join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000              0 row(s) returned       0.000 sec / 0.000 sec
60  23:10:24  select * from likes left join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000         1000 row(s) returned    0.063 sec / 0.000 sec
61  23:11:04  select * from likes left join users on likes.user_id=users.id ORDER BY photo_id desc LIMIT 0, 1000               1000 row(s) returned    0.015 sec / 0.016 sec
```

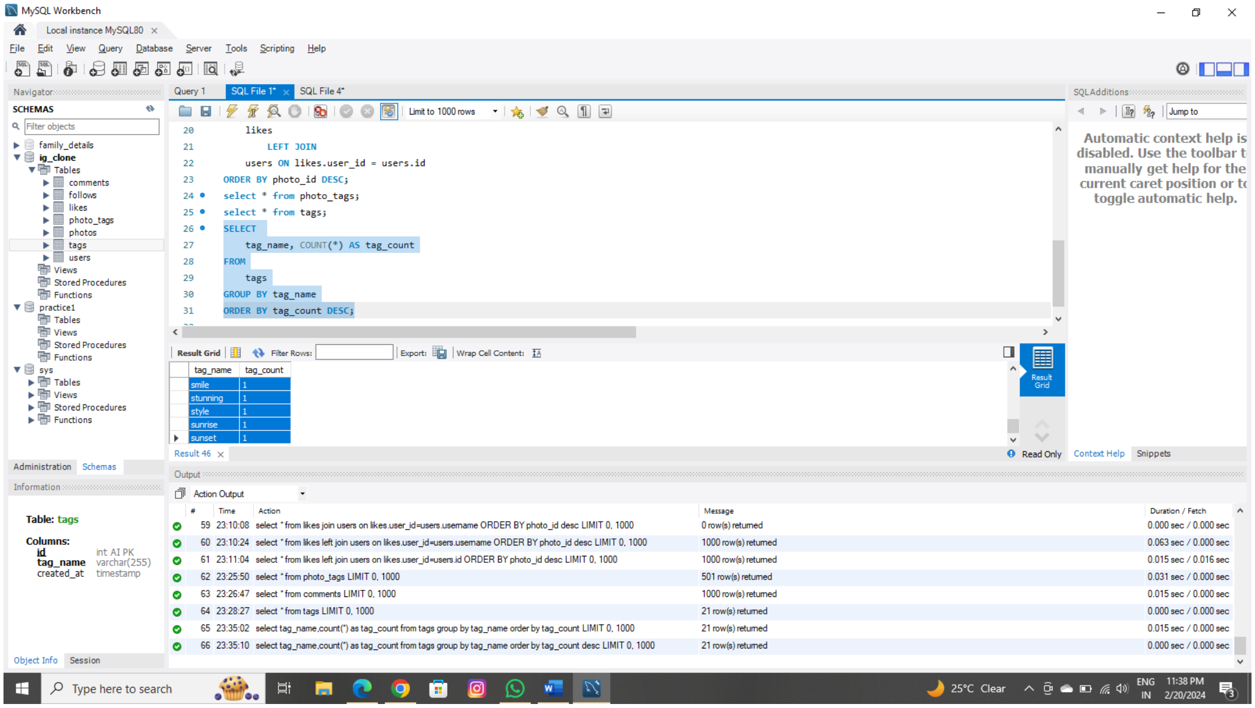
Question 4; Hash tag research

Identifying and suggesting the top five most commonly used hashtags on the platform;

From the provided database we have obtained that these are the most and top 5 commonly used hashtags on the platform;

- Since all the hash tags are different we have all count of them are 1, but if we observe the results we have similarities of **food,fun and happy, fashion and style.**

Hash_tag_names	count
beach	1
beauty	1
concert	1
delicious	1
dreamy	1
drunk	1
fashion	1
food	1
foodie	1
fun	1
hair	1
happy	1
landscape	1
lol	1
party	1
photography	1
smile	1
stunning	1
style	1
sunrise	1
sunset	1



```
        likes
            LEFT JOIN
        users ON likes.user_id = users.id
    ORDER BY photo_id DESC;
    select * from photo_tags;
    select * from tags;
    SELECT
        tag_name, COUNT(*) AS tag_count
    FROM
        tags
    GROUP BY tag_name
    ORDER BY tag_count DESC;
```

tag_name	tag_count
smile	1
stunning	1
style	1
sunrise	1
sunset	1

Table: tags

Columns:

id	int AI PK
tag_name	varchar(255)
created_at	timestamp

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

```
#   Time      Action                                                                                        Message              Duration / Fetch
59  23:10:08  select * from likes join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000        0 row(s) returned     0.000 sec / 0.000 sec
60  23:10:24  select * from likes left join users on likes.user_id=users.username ORDER BY photo_id desc LIMIT 0, 1000   1000 row(s) returned  0.063 sec / 0.000 sec
61  23:11:04  select * from likes left join users on likes.user_id=users.id ORDER BY photo_id desc LIMIT 0, 1000         1000 row(s) returned  0.015 sec / 0.016 sec
62  23:25:50  select * from photo_tags LIMIT 0, 1000                                                                    501 row(s) returned   0.031 sec / 0.000 sec
63  23:26:47  select * from comments LIMIT 0, 1000                                                                      1000 row(s) returned  0.015 sec / 0.000 sec
64  23:28:27  select * from tags LIMIT 0, 1000                                                                          21 row(s) returned    0.000 sec / 0.000 sec
65  23:35:02  select tag_name,count(*) as tag_count from tags group by tag_name order by tag_count LIMIT 0, 1000        21 row(s) returned    0.015 sec / 0.000 sec
66  23:35:10  select tag_name,count(*) as tag_count from tags group by tag_name order by tag_count desc LIMIT 0, 1000   21 row(s) returned    0.000 sec / 0.000 sec
```

- Above is the screen shot of the queries has been ran.

Question 5; Ad campaign launch.

Determining the day of the week when most users register on Instagram. to provide insights on when to schedule an ad campaign.

From the provided database we have obtained that these are the most and top 5 commonly used hashtags on the platform;

No of Registrations	On the day of a week
16	Thursday
16	Sunday
15	Friday
14	Tuesday

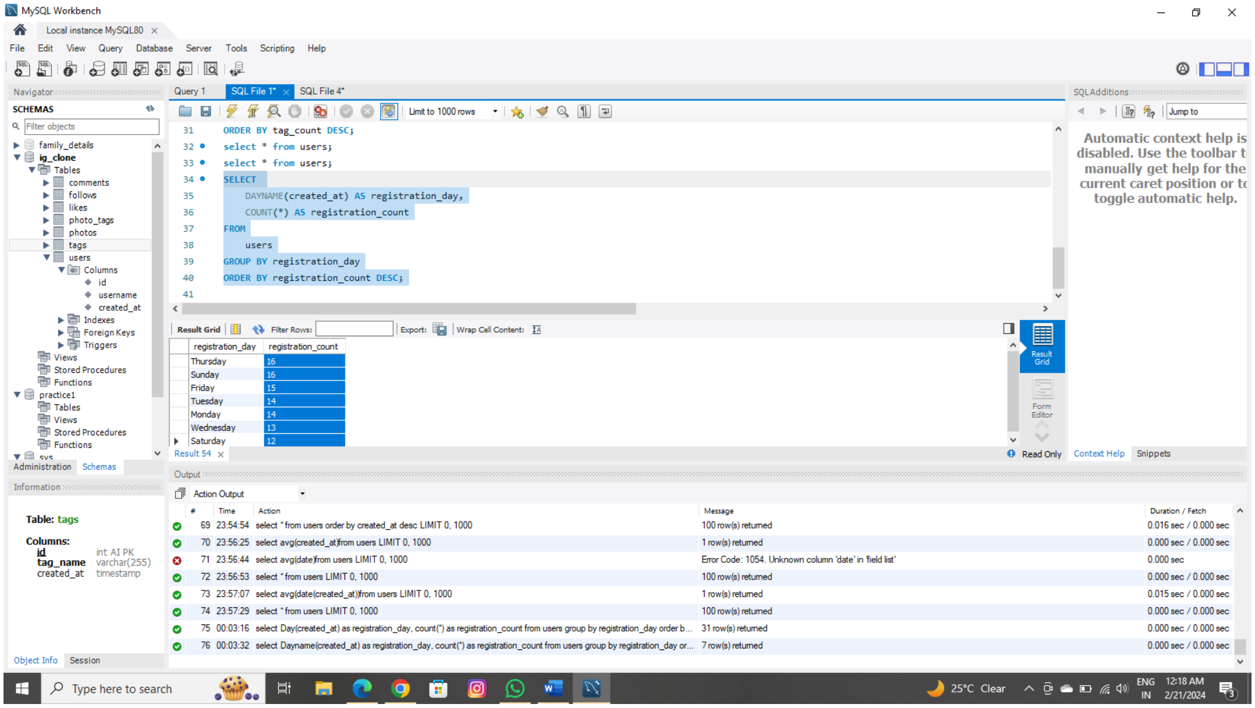
14	Monday
13	Wednesday
12	Saturday

- Since we have most user registrations on the days Thursday and Sunday compared to other week days it's good to go on Thursdays and Sundays especially. The ad campaign may give good results if we run ads on particularly those days.

- These are the queries had been written to get results.
- Here, we are used a special function before selecting the date called "**datename"** to get the name of the day instead of the day number

```
SELECT
   DAYNAME(created_at) AS registration_day,
   COUNT(*) AS registration_count
FROM
   users
GROUP BY registration_day
ORDER BY registration_count DESC;
```

- Here we have snapshot of the workbench



```
ORDER BY tag_count DESC;
select * from users;
select * from users;
SELECT
    DAYNAME(created_at) AS registration_day,
    COUNT(*) AS registration_count
FROM
    users
GROUP BY registration_day
ORDER BY registration_count DESC;
```

registration_day	registration_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Table: tags

Columns:

id	int AI PK
tag_name	varchar(255)
created_at	timestamp

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#   Time      Action                                                                                         Message                                               Duration / Fetch
69  23:54:54  select * from users order by created_at desc LIMIT 0, 1000                                     100 row(s) returned                                   0.016 sec / 0.000 sec
70  23:56:25  select avg(created_at)from users LIMIT 0, 1000                                                 1 row(s) returned                                     0.000 sec / 0.000 sec
71  23:56:44  select avg(date)from users LIMIT 0, 1000                                                       Error Code: 1054. Unknown column 'date' in 'field list'  0.000 sec
72  23:56:53  select * from users LIMIT 0, 1000                                                              100 row(s) returned                                   0.000 sec / 0.000 sec
73  23:57:07  select avg(date(created_at))from users LIMIT 0, 1000                                           1 row(s) returned                                     0.015 sec / 0.000 sec
74  23:57:29  select * from users LIMIT 0, 1000                                                              100 row(s) returned                                   0.000 sec / 0.000 sec
75  00:03:16  select Day(created_at) as registration_day, count(*) as registration_count from users group by registration_day order b...  31 row(s) returned  0.000 sec / 0.000 sec
76  00:03:32  select Dayname(created_at) as registration_day, count(*) as registration_count from users group by registration_day or...  7 row(s) returned  0.000 sec / 0.000 sec
```

Task B: Investor metrics;

Question 1;

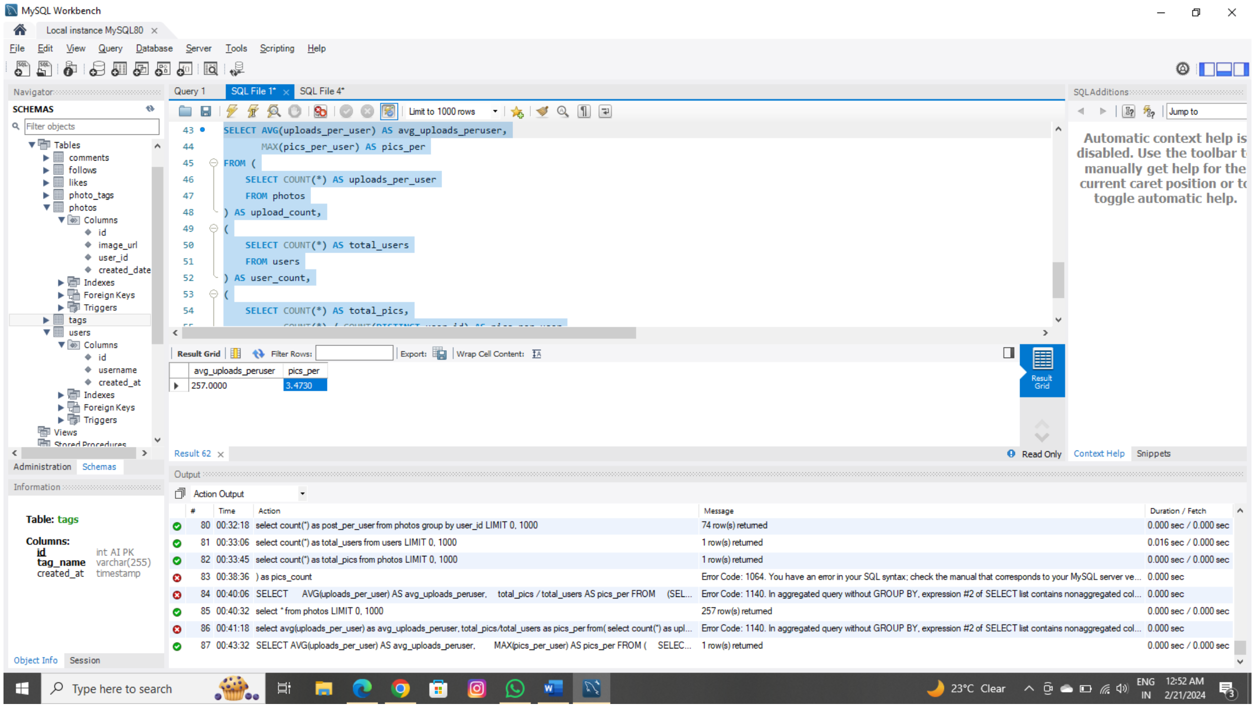
Calculating the average number of the posts per user on Instagram and providing the total number of photos on Instagram divide by the total number of users.

From the provided database we have obtained that these are the average number of the posts per user on Instagram and the total number of photos on Instagram divided by the total number of users;

- ✓ Average uploads per user are **257.0000**
- ✓ Pics per user **3.4730 (** approximately **4 pics per user)**
- ❖ Here the query that had been ran by the workbench to get the results

```
SELECT AVG(uploads_per_user) AS avg_uploads_peruser,
       MAX(pics_per_user) AS pics_per
FROM (
   SELECT COUNT(*) AS uploads_per_user
   FROM photos
) AS upload_count,
(
   SELECT COUNT(*) AS total_users
   FROM users
) AS user_count,
(
   SELECT COUNT(*) AS total_pics,
          COUNT(*) / COUNT(DISTINCT user_id) AS pics_per_user
   FROM photos
) AS pics_count;
```

- This is snapshot of the workbench without put



```
SELECT AVG(uploads_per_user) AS avg_uploads_peruser,
       MAX(pics_per_user) AS pics_per
FROM (
    SELECT COUNT(*) AS uploads_per_user
    FROM photos
) AS upload_count,
(
    SELECT COUNT(*) AS total_users
    FROM users
) AS user_count,
(
    SELECT COUNT(*) AS total_pics,
```

	avg_uploads_peruser	pics_per
▶	257.0000	3.4730

```
#   Time      Action                                                                                        Message                                                                                           Duration / Fetch
80  00:32:18  select count(*) as post_per_user from photos group by user_id LIMIT 0, 1000                    74 row(s) returned                                                                                0.000 sec / 0.000 sec
81  00:33:06  select count(*) as total_users from users LIMIT 0, 1000                                       1 row(s) returned                                                                                 0.016 sec / 0.000 sec
82  00:33:45  select count(*) as total_pics from photos LIMIT 0, 1000                                       1 row(s) returned                                                                                 0.000 sec / 0.000 sec
83  00:38:36  ) as pics_count                                                                               Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ve...   0.000 sec
84  00:40:06  SELECT    AVG(uploads_per_user) AS avg_uploads_peruser,    total_pics / total_users AS pics_per FROM    (SEL...   Error Code: 1140. In aggregated query without GROUP BY, expression #2 of SELECT list contains nonaggregated col...   0.000 sec
85  00:40:32  select * from photos LIMIT 0, 1000                                                            257 row(s) returned                                                                               0.000 sec / 0.000 sec
86  00:41:18  select avg(uploads_per_user) as avg_uploads_peruser, total_pics/total_users as pics_per from( select count(*) as upl...   Error Code: 1140. In aggregated query without GROUP BY, expression #2 of SELECT list contains nonaggregated col...   0.000 sec
87  00:43:32  SELECT AVG(uploads_per_user) AS avg_uploads_peruser,    MAX(pics_per_user) AS pics_per FROM (    SELEC...   1 row(s) returned                                                                                 0.000 sec / 0.000 sec
```

Question 2: Bots & Fake Accounts

Identifying users who have liked every single photo on the site, as it’s not typically possible for a normal user.

From the provided database we have obtained that these are the users who has liked every single photo on the site.

User_id’s	No of likes
21	257
71	257
5	257
66	257
41	257

14	257
57	257
24	257
76	257
75	257
54	257
91	257
36	257

- these are the queries has been run to get those results..

query 1:

```
select count(image_url) from photos;
```

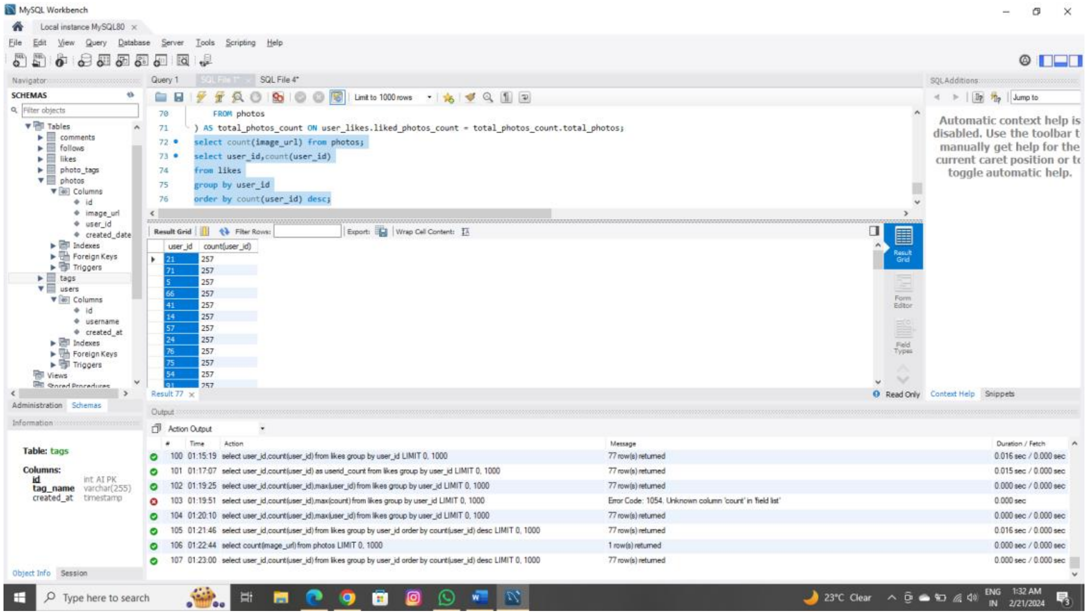
query 2:

```
select user_id,count(user_id)
from likes
group by user_id
order by count(user_id) desc;
```

in the above query's we have first ran the query to get how many photos are there totally and after we have conclude that who has the likes those are equal to the total numbers of photos those Id's will be bots or fake accounts.

(total photos = 257)

This is the screenshot of the workbench….



```
        FROM photos
) AS total_photos_count ON user_likes.liked_photos_count = total_photos_count.total_photos;
select count(image_url) from photos;
select user_id,count(user_id)
from likes
group by user_id
order by count(user_id) desc;
```

user_id	count(user_id)
21	257
71	257
5	257
66	257
41	257
14	257
57	257
24	257
76	257
75	257
54	257
91	257

#	Time	Action	Message	Duration / Fetch
100	01:15:19	select user_id,count(user_id) from likes group by user_id LIMIT 0, 1000	77 row(s) returned	0.016 sec / 0.000 sec
101	01:17:07	select user_id,count(user_id) as userid_count from likes group by user_id LIMIT 0, 1000	77 row(s) returned	0.015 sec / 0.000 sec
102	01:19:25	select user_id,count(user_id),max(user_id) from likes group by user_id LIMIT 0, 1000	77 row(s) returned	0.000 sec / 0.000 sec
103	01:19:51	select user_id,count(user_id),max(count) from likes group by user_id LIMIT 0, 1000	Error Code: 1054. Unknown column 'count' in 'field list'	0.000 sec
104	01:20:10	select user_id,count(user_id),max(user_id) from likes group by user_id LIMIT 0, 1000	77 row(s) returned	0.000 sec / 0.000 sec
105	01:21:46	select user_id,count(user_id) from likes group by user_id order by count(user_id) desc LIMIT 0, 1000	77 row(s) returned	0.016 sec / 0.000 sec
106	01:22:44	select count(image_url) from photos LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
107	01:23:00	select user_id,count(user_id) from likes group by user_id order by count(user_id) desc LIMIT 0, 1000	77 row(s) returned	0.000 sec / 0.000 sec

- in all above information we have answered all the questions

✓ Loyal user reward
✓ Inactive user engagement
✓ Contest winner declaration
✓ Hashtag research
✓ Ad campaign

From task B

✓ User engagement
✓ Bots and fake account detection

3. Tech-stack used

- In this project I have used the MySQL workbench software.
- Server have been used SQL community Server – GPL
- Version of the MySQL 8.0.36
- Windows 10, 8 gb ram, 256 ssd

I’ve chosen the setup environment and the configuration because

Data Integrity: Ensures accuracy and consistency of data through constraints and relationships.
Scalability: Handles large volumes of data efficiently, supporting thousands to millions of users.
Query flexibility: Empowers complex data retrieval, analysis, and manipulation with powerful querying capabilities.

- And my SQL workbench is the free and user-friendly setup to practice and understand the SQL queries.

4. Insights

- I've got some useful insights and patterns across the users and their behavior towards the Instagram app
- I've resolved many questions with answers with the use of data.
- And especially the things which are I solved from the problems these are some of my favorites like "Detecting the inactive users and finding best day to run ad campaign and also finding fake or bots used by users in the app". Those are very interested to me about working on them.

5. Results

- I've made a wonderful achievement throughout this project I've learned how to use and where to use and when to use the queries based on situations
- I came to know in entire project how it is a data analysis take places in every department using the data to get useful insights and patterns and trends over the time.
- All these results will use to solve the problems and get more profits through the business.