



INSTAGRAM

USER ANALYTICS

TRAINITY
PROJECT-2

DESCRIPTION

Welcome to the data analysis project, where I delve into the intricate world of Instagram(clone).As a learning data analyst embedded with the Instagram trainity team, I'm diving into user interaction metrics to uncover valuable insights that can shape the future of this iconic social media platform. Using the powerful combination of SQL and MySQL Workbench, my mission is to translate data into strategic decisions, from boosting user engagement to refining product features, and empower the product, marketing and investor teams with data-driven intelligence .

By analyzing:

- Most loyal users
- Inactive users
- User with the most likes on a single post
- most popular hashtags
- Day of the week when most users registered on Instagram
- Ad campaign launch
- Boats and Fake accounts

APPROACH:

- Firstly, I imported the dataset provided by team into MySQL workbench
- Understood the Data schema thoroughly ,the structure of Instagram database ,including tables, columns and relationship
- Data analysis using MYSQL
- Applying appropriate queries to solve given problem statements

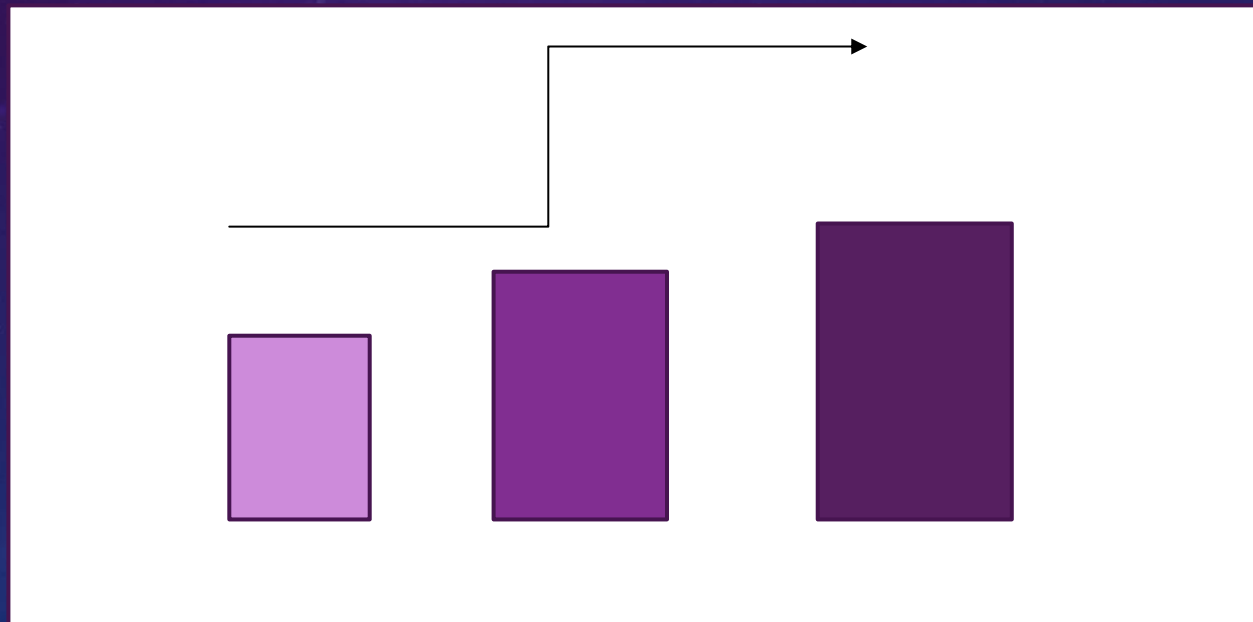
TECH-STACK USED:

- **MYSQL WORKBENCH 8.0 CE**

MySQL Workbench 8.0 CE offers a user-friendly interface for managing databases, featuring tools for visual database design, SQL development, and administration. It allows users to create and manage ER diagrams, write and debug SQL code, and handle tasks like backups, recovery, and user management. With cross-platform compatibility, visual performance dashboards, and support for plugins and scripting, it simplifies database design, development, and maintenance, making it an ideal tool for developers and administrators.

INSIGHTS

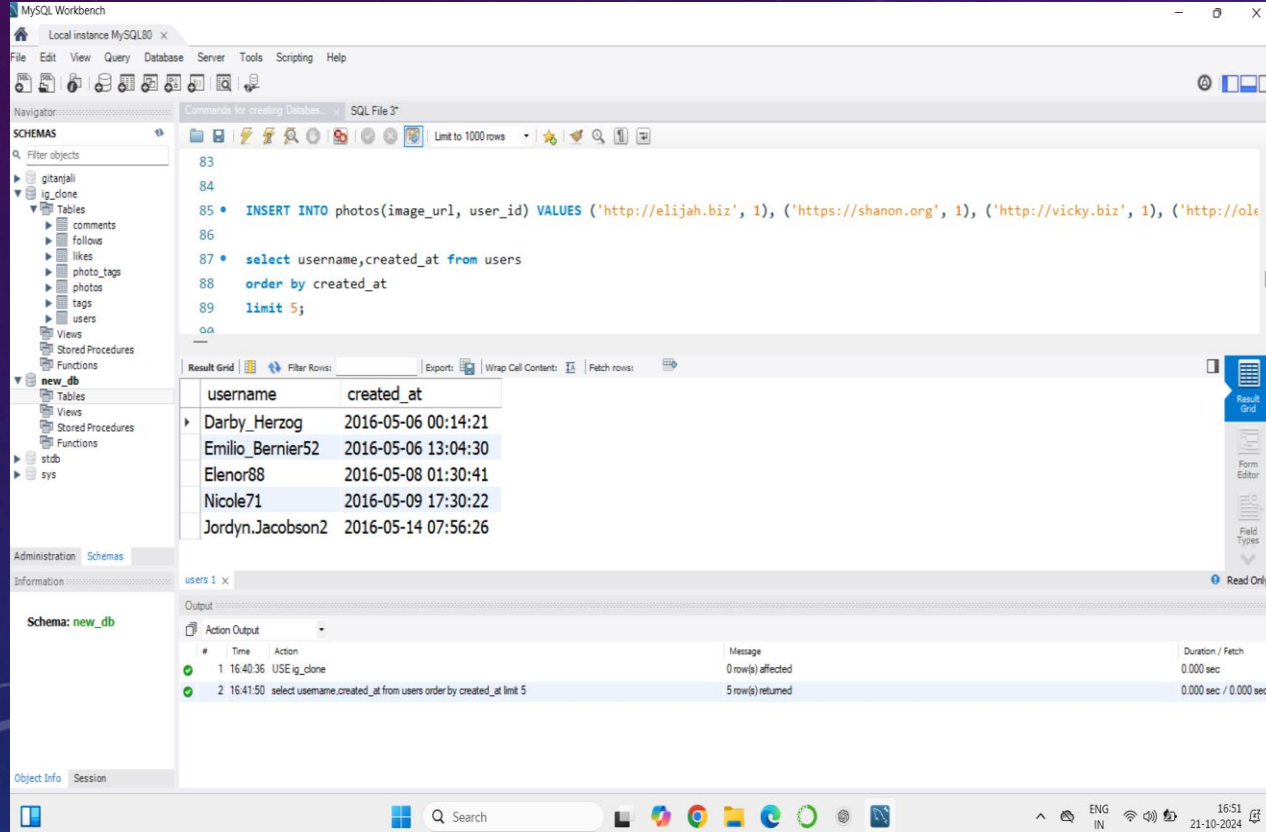
Marketing Analysis



1. LOYAL USER REWARD

TASK: The top five oldest users on Instagram from the provided dataset.

The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
83  
84  
85 • INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1), ('http://ole  
86  
87 • select username,created_at from users  
88 order by created_at  
89 limit 5;
```

The Results window displays the following data:

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

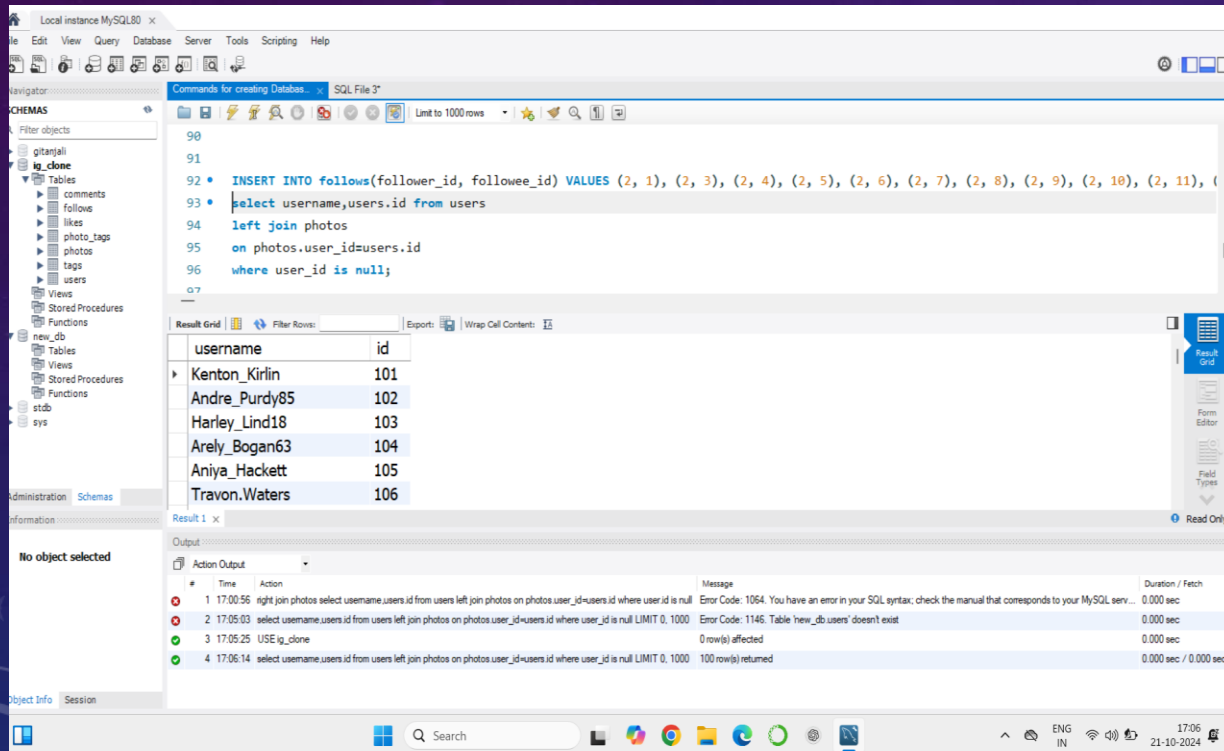
The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	16:40:36	USE ig_clone	0 row(s) affected	0.000 sec
2	16:41:50	select username,created_at from users order by created_at limit 5	5 row(s) returned	0.000 sec / 0.000 sec

- To find the solution to this problem, I simply selected the required the columns from users table and ordered them accordingly. Since the required limit was 5, so I put on the limit condition to be 5.

2. ACTIVE USER ENGAGEMENT

Task: Identify users who have never posted a single photo on Instagram



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
90
91
92 • INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10), (2, 11), (
93 • select username,users.id from users
94 left join photos
95 on photos.user_id=users.id
96 where user_id is null;
97
```

The Results Grid shows the following data:

username	id
Kenton_Kirlin	101
Andre_Purdy85	102
Harley_Lind18	103
Arely_Bogan63	104
Aniya_Hackett	105
Travon.Waters	106

The Action Output pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
1	17:00:56	right join photos select username,users id from users left join photos on photos.user_id=users id where user_id is null	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...	0.000 sec
2	17:05:03	select username,users id from users left join photos on photos.user_id=users id where user_id is null LIMIT 0, 1000	Error Code: 1146. Table 'new_db.users' doesn't exist	0.000 sec
3	17:05:25	USE iq_clone	0 row(s) affected	0.000 sec
4	17:06:14	select username,users id from users left join photos on photos.user_id=users id where user_id is null LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

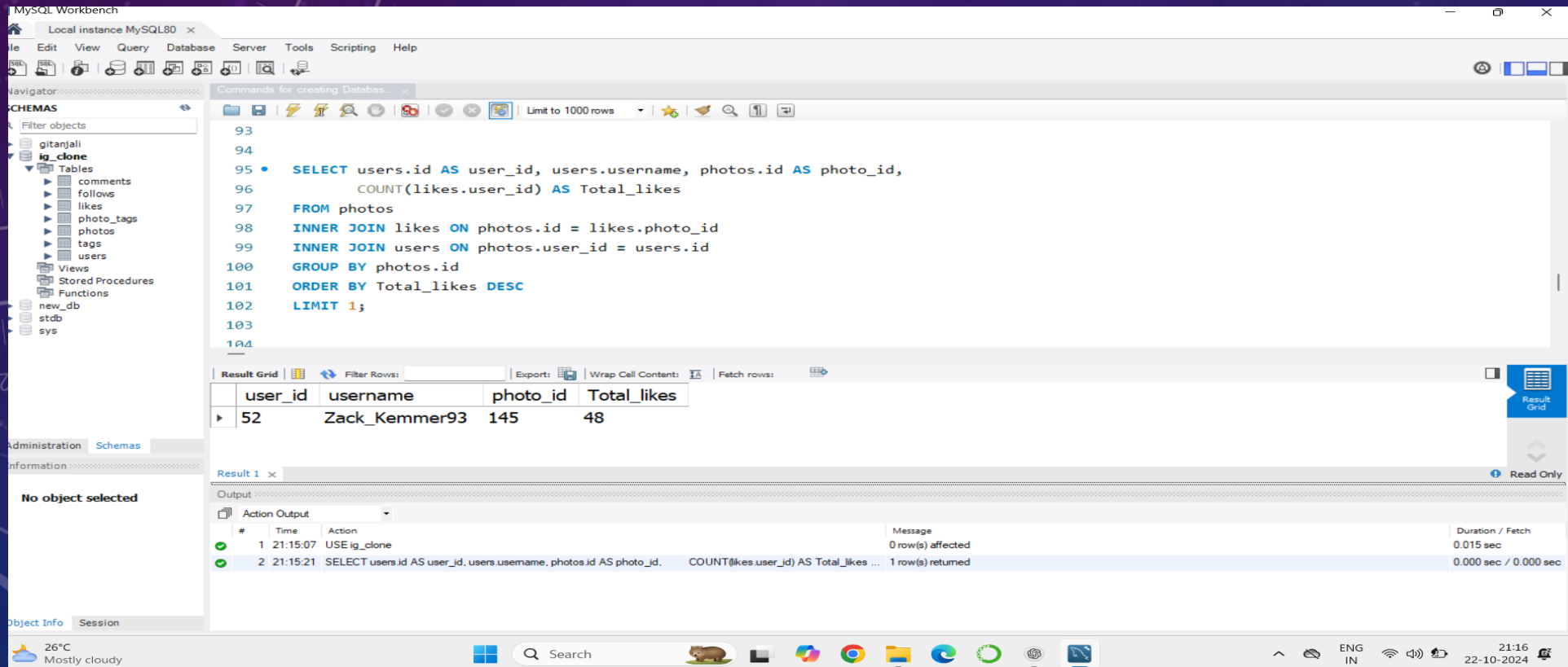
The team wants to encourage inactive users to start posting by sending them promotional emails.

To solve this problem, I joined two tables namely, users and photos. After joining, I applied 'where' condition to identify the users who never posted a photo.

3. CONTEST WINNER DECLARATION

TASK: determine the winner of the contest and provide their details to the team

- The user id having 52 with the most likes on a single post wins the contest



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view of databases including 'gitanjali', 'ig_clone', 'new_db', 'stdb', and 'sys'. The 'ig_clone' database is selected, showing tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', and 'users'. The main editor window contains a SQL query:

```
93
94
95 • SELECT users.id AS user_id, users.username, photos.id AS photo_id,
96       COUNT(likes.user_id) AS Total_likes
97 FROM photos
98 INNER JOIN likes ON photos.id = likes.photo_id
99 INNER JOIN users ON photos.user_id = users.id
100 GROUP BY photos.id
101 ORDER BY Total_likes DESC
102 LIMIT 1;
103
104
```

Below the query editor, the 'Result Grid' shows the output of the query:

user_id	username	photo_id	Total_likes
52	Zack_Kemmer93	145	48

The bottom panel shows the 'Action Output' log with the following entries:

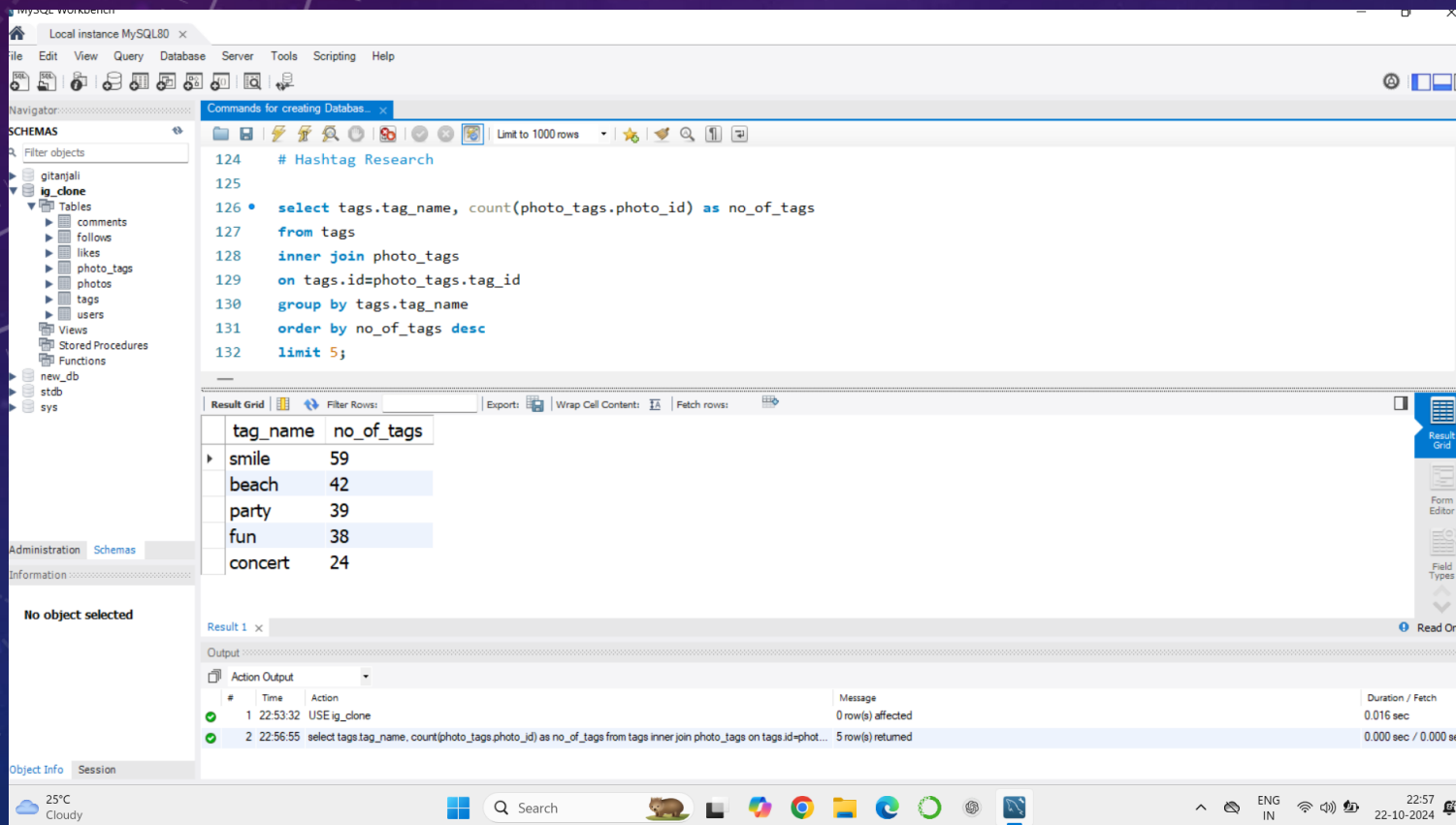
#	Time	Action	Message	Duration / Fetch
1	21:15:07	USE ig_clone	0 row(s) affected	0.015 sec
2	21:15:21	SELECT users.id AS user_id, users.username, photos.id AS photo_id, COUNT(likes.user_id) AS Total_likes ...	1 row(s) returned	0.000 sec / 0.000 sec

The Windows taskbar at the bottom shows the system clock as 21:16 on 22-10-2024, with a temperature of 26°C and weather 'Mostly cloudy'.

4. HASHTAG RESEARCH

TASK: identify and suggest the top five most commonly used hashtags on the platform

The top five most commonly used hashtag to use in their posts to reach the most peoples are : smile, beach, party, fun, concert .



The screenshot displays the MySQL Workbench interface. The 'Schemas' panel on the left shows a database named 'ig_clone' with various tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', and 'users'. The main editor window contains a SQL query titled '# Hashtag Research' which uses an inner join to count the number of photos for each tag. The 'Result Grid' shows the top five results: 'smile' (59), 'beach' (42), 'party' (39), 'fun' (38), and 'concert' (24). The 'Output' panel at the bottom shows the execution of the query, indicating that 5 rows were returned.

```
124 # Hashtag Research
125
126 • select tags.tag_name, count(photo_tags.photo_id) as no_of_tags
127 from tags
128 inner join photo_tags
129 on tags.id=photo_tags.tag_id
130 group by tags.tag_name
131 order by no_of_tags desc
132 limit 5;
```

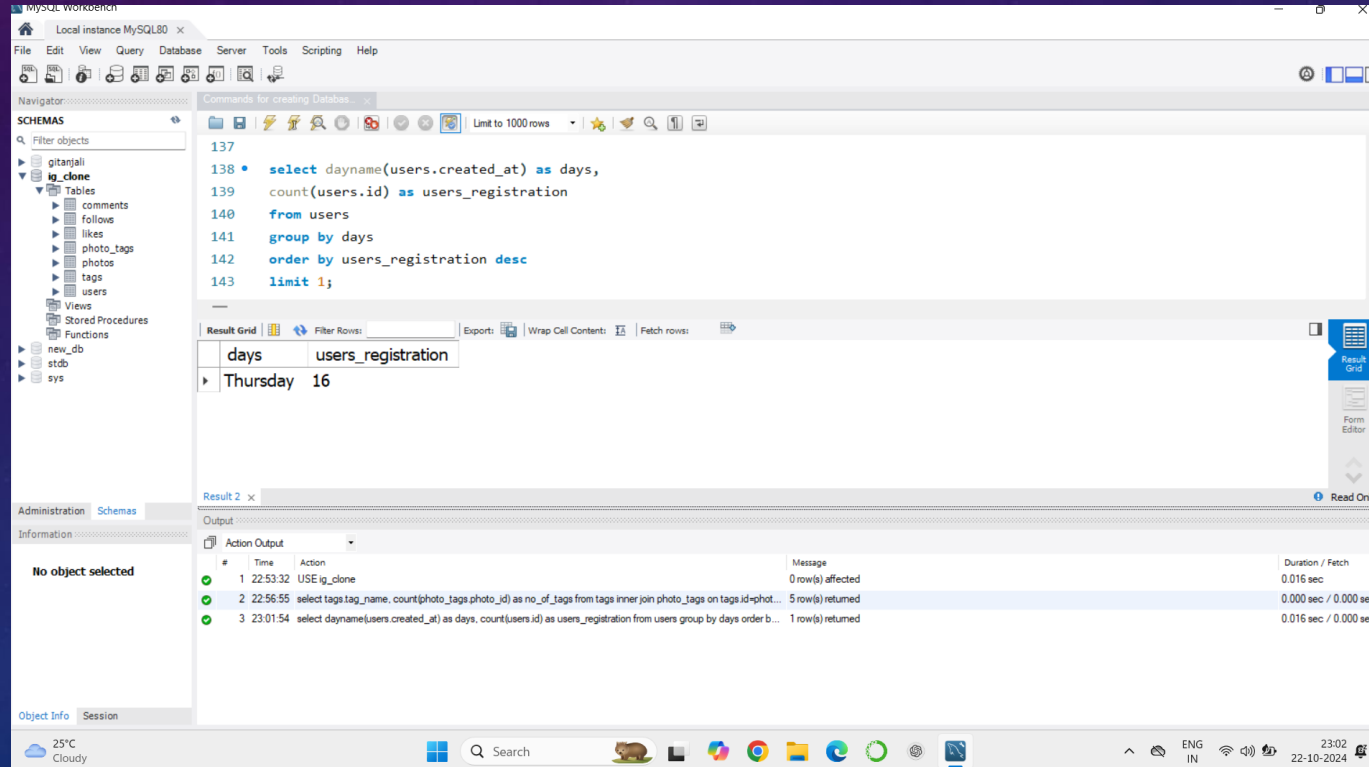
tag_name	no_of_tags
smile	59
beach	42
party	39
fun	38
concert	24

#	Time	Action	Message	Duration / Fetch
1	22:53:32	USE ig_clone	0 row(s) affected	0.016 sec
2	22:56:55	select tags.tag_name, count(photo_tags.photo_id) as no_of_tags from tags inner join photo_tags on tags.id=photo_tags.tag_id group by tags.tag_name order by no_of_tags desc limit 5;	5 row(s) returned	0.000 sec / 0.000 sec

5. AD CAMPAIGN LAUNCH

Task: Determine the day of the week when most users registered on Instagram

The best day of the week to launch ads is : THURSDAY



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view containing 'gitanjali', 'ig_clone' (selected), 'new_db', 'stdb', and 'sys'. The 'ig_clone' schema is expanded, showing tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', and 'users'. The main editor window contains a SQL query:

```
137
138 • select dayname(users.created_at) as days,
139       count(users.id) as users_registration
140 from users
141 group by days
142 order by users_registration desc
143 limit 1;
```

Below the query editor, the 'Result Grid' shows the output of the query:

days	users_registration
Thursday	16

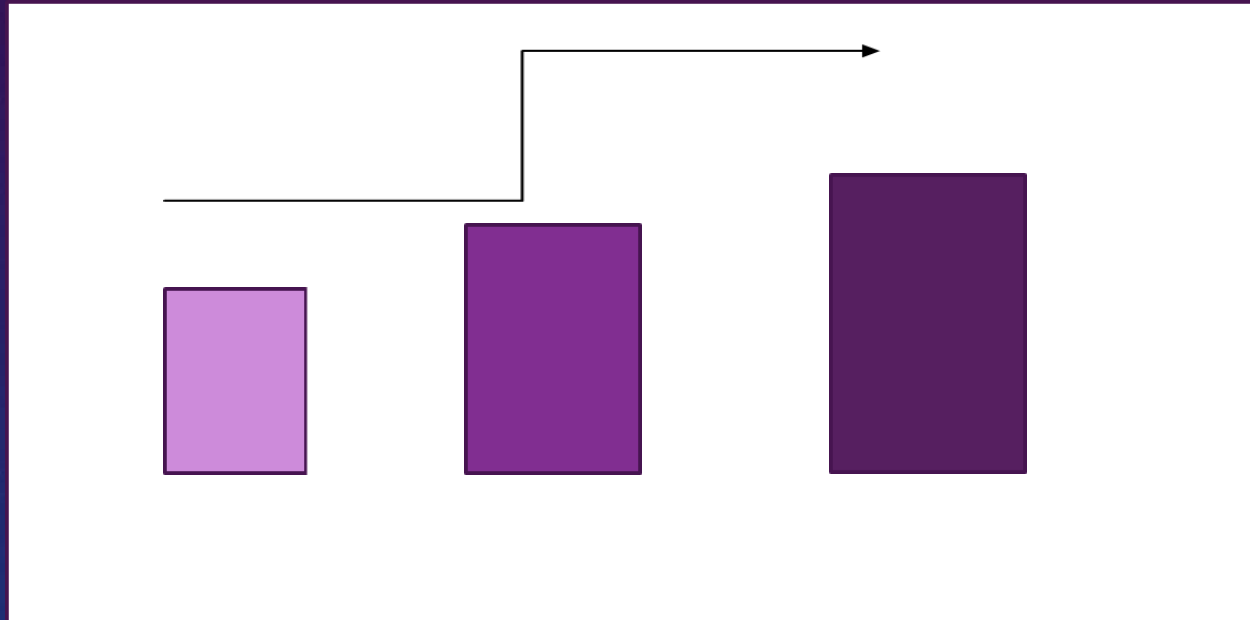
The bottom panel shows the 'Output' tab with a table of action messages:

#	Time	Action	Message	Duration / Fetch
1	22:53:32	USE ig_clone	0 row(s) affected	0.016 sec
2	22:56:55	select tags.tag_name, count(photo_tags.photo_id) as no_of_tags from tags inner join photo_tags on tags.id=phot...	5 row(s) returned	0.000 sec / 0.000 sec
3	23:01:54	select dayname(users.created_at) as days, count(users.id) as users_registration from users group by days order b...	1 row(s) returned	0.016 sec / 0.000 sec

The bottom status bar shows the system clock as 23:02 on 22-10-2024, along with other system icons.

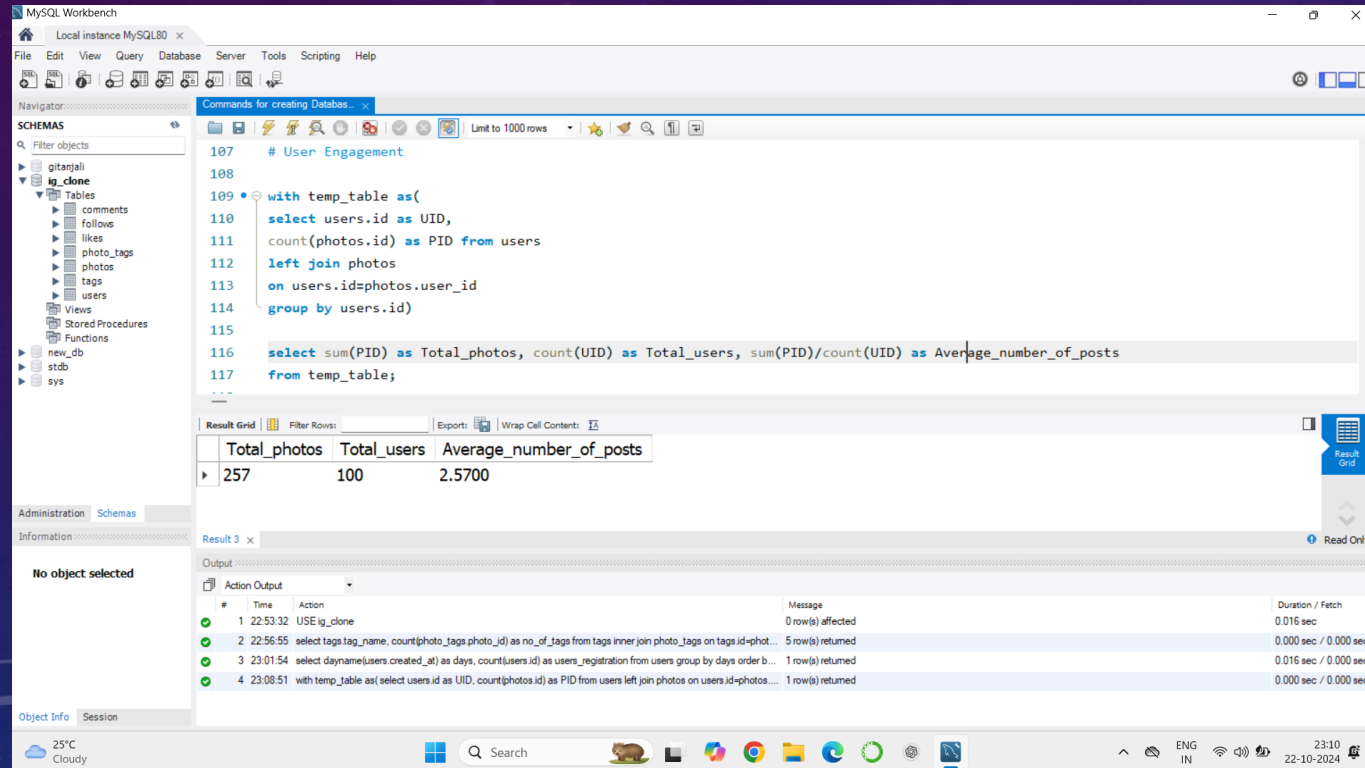
INSIGHTS

Investor metrics



6. USER ENGAGEMENT

Task: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
107 # User Engagement
108
109 with temp_table as(
110 select users.id as UID,
111 count(photos.id) as PID from users
112 left join photos
113 on users.id=photos.user_id
114 group by users.id)
115
116 select sum(PID) as Total_photos, count(UID) as Total_users, sum(PID)/count(UID) as Average_number_of_posts
117 from temp_table;
```

The Results Grid shows the following data:

Total_photos	Total_users	Average_number_of_posts
257	100	2.5700

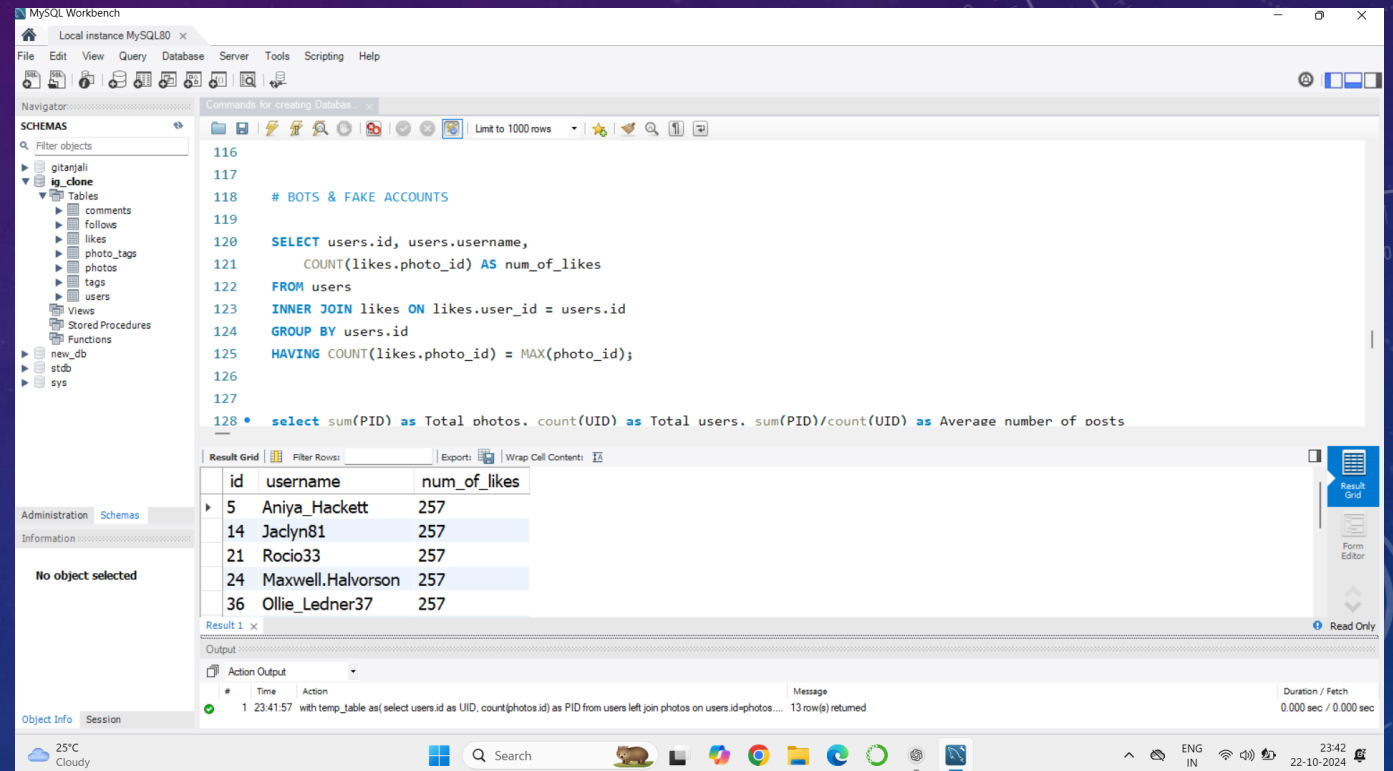
The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	22:53:32	USE ig_clone	0 row(s) affected	0.016 sec
2	22:56:55	select tags.tag_name, count(photo_tags.photo_id) as no_of_tags from tags inner join photo_tags on tags.id=photo_tags.tag_id	5 row(s) returned	0.000 sec / 0.000 sec
3	23:01:54	select dayname(users.created_at) as days, count(users.id) as users_registration from users group by days order by days	1 row(s) returned	0.016 sec / 0.000 sec
4	23:08:51	with temp_table as(select users.id as UID, count(photos.id) as PID from users left join photos on users.id=photos.user_id group by users.id)	1 row(s) returned	0.000 sec / 0.000 sec

Total_photos : 257
Total_users : 100
Average_no_of_posts: 2.5700

7. BOTS AND FAKE ACCOUNTS

TASK: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a filter on 'ig_clone'. The main editor window contains a SQL query to identify potential bot accounts. The query is as follows:

```
116
117
118 # BOTS & FAKE ACCOUNTS
119
120 SELECT users.id, users.username,
121        COUNT(likes.photo_id) AS num_of_likes
122 FROM users
123 INNER JOIN likes ON likes.user_id = users.id
124 GROUP BY users.id
125 HAVING COUNT(likes.photo_id) = MAX(photo_id);
126
127
128 • select sum(PID) as Total photos, count(UID) as Total users, sum(PID)/count(UID) as Average number of posts
```

The 'Result Grid' shows the following data:

id	username	num_of_likes
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257

The 'Output' pane at the bottom shows the execution message: '1 23:41:57 with temp_table as select users.id as UID, count(photos.id) as PID from users left join photos on users.id=photos... 13 row(s) returned'. The status bar at the bottom indicates the system is at 25°C, Cloudy, and the time is 23:42 on 22-10-2024.

RESULT

- FOR MARKETING TEAM:
 1. The 5 most loyal users identified.
 2. There are 26 inactive users who have never posted a single post.
 3. Zack_kimmer93 is the winner of contest.
 4. The top five commonly used hashtags identified.
 5. THURSDAY IS THE BEST DAY TO REGEDTER ON INSTAGRAM.
- FOR INVESTOR TEAM:
 1. The average number of posts per user is 2-3 in a day.
 2. There are 23 account found to be fake as they liked all 257 posts.

- I completed this project by learning some basic lectures of MySQL from TRAINITY platform. This project helped me to work on real-life-problems and gained practical knowledge while working on this project and learned to apply most of the Functionalities. From this project I got to know how the real-life problems tackled to get valuable insights for business which would contribute to the growth of companies/organization.

THANKS!

GITANJALI PEKAMWAR