

# TRAINITY PROJECT 2: DATA ANALYTICS

## PROCESS

### TOPIC: Instagram User Analytics Using MySQL Workbench

#### PROJECT DESCRIPTION:

As a data analyst working closely with the product team at Instagram, my primary responsibility is to explore and analyze user behaviors and engagement on the Instagram app. These analyses are vital for formulating data-driven decisions that further the growth and success of the platform. Utilizing SQL and MySQL Workbench as my tools, I strive to uncover valuable insights from Instagram user data, which empowers numerous teams across the business.

This project aims to examine user engagement on Instagram and offer valuable insights that can aid the product team in refining the platform. By thoroughly analyzing user interactions, including likes, comments, follows, and posts, we can gain a deeper comprehension of user behaviors and preferences. This knowledge can be leveraged by various teams, encompassing marketing, product development, and customer support, to optimize the user experience and impel business growth.

#### APPROACH:

**Create a Database:** The database file was provided in the attachments.

```
1 • CREATE DATABASE ig_clone;
2 • USE ig_clone;
3
4 /*Users*/
5 • CREATE TABLE users(
6     id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
7     username VARCHAR(255) NOT NULL,
8     created_at TIMESTAMP DEFAULT NOW()
9 );
10
11 /*Photos*/
12 • CREATE TABLE photos(
13     id INT AUTO_INCREMENT PRIMARY KEY,
14     image_url VARCHAR(355) NOT NULL,
15     user_id INT NOT NULL,
16     created_dat TIMESTAMP DEFAULT NOW(),
17     FOREIGN KEY(user_id) REFERENCES users(id)
18 );
```

```
20      /*Comments*/
21  ● ○ CREATE TABLE comments(
22      id INT AUTO_INCREMENT PRIMARY KEY,
23      comment_text VARCHAR(255) NOT NULL,
24      user_id INT NOT NULL,
25      photo_id INT NOT NULL,
26      created_at TIMESTAMP DEFAULT NOW(),
27      FOREIGN KEY(user_id) REFERENCES users(id),
28      FOREIGN KEY(photo_id) REFERENCES photos(id)
29  );
30
31      /*Likes*/
32  ● ○ CREATE TABLE likes(
33      user_id INT NOT NULL,
34      photo_id INT NOT NULL,
35      created_at TIMESTAMP DEFAULT NOW(),
36      FOREIGN KEY(user_id) REFERENCES users(id),
37      FOREIGN KEY(photo_id) REFERENCES photos(id),
38      PRIMARY KEY(user_id,photo_id)
39  );
40
41      /*follows*/
42  ● ○ CREATE TABLE follows(
43      follower_id INT NOT NULL,
44      followee_id INT NOT NULL,
45      created_at TIMESTAMP DEFAULT NOW(),
46      FOREIGN KEY (follower_id) REFERENCES users(id),
47      FOREIGN KEY (followee_id) REFERENCES users(id),
48      PRIMARY KEY(follower_id,followee_id)
49  );
50
51      /*Tags*/
52  ● ○ CREATE TABLE tags(
53      id INTEGER AUTO_INCREMENT PRIMARY KEY,
54      tag_name VARCHAR(255) UNIQUE NOT NULL,
55      created_at TIMESTAMP DEFAULT NOW()
56  );
57
```

---

```

58      /*junction table: Photos - Tags*/
59  ● CREATE TABLE photo_tags(
60      photo_id INT NOT NULL,
61      tag_id INT NOT NULL,
62      FOREIGN KEY(photo_id) REFERENCES photos(id),
63      FOREIGN KEY(tag_id) REFERENCES tags(id),
64      PRIMARY KEY(photo_id,tag_id)
65  );
66
67  ● SELECT * FROM users;
68  ● SELECT * FROM photos;
69  ● SELECT * FROM comments;
70  ● SELECT * FROM likes;
71  ● SELECT * FROM follows;
72  ● SELECT * FROM tags;
73  ● SELECT * FROM photo_tags;
74

```

**Perform Analysis:** I utilised MySQL Workbench to perform the analysis and answer the questions mentioned in the project description. I used SQL queries to extract the required information from the database, ensuring the accuracy and efficiency of the queries.

### **TECH- STACK USED:**

The software I am using to do the analysis is MySQL Workbench, Version: 8.0.34 (MySQL Community Server – GPL). I used this software for the project as it is easy to use and freely available.

### **INSIGHTS:**

#### **A) Marketing Analysis:**

1. **Loyal User Reward:** The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

To identify the five oldest users on Instagram from the provided database:

```

SELECT id, username FROM users
ORDER BY created_at ASC
LIMIT 5;

```

The SQL query provided retrieves rows from the **users** table, orders them by the **created\_at** column in ascending order (earliest to latest), and limits the result to the first 5 rows.

Result Grid			Filter Rows:
	id	username	
▶	80	Darby_Herzog	
	67	Emilio_Bernier52	
	63	Elenor88	
	95	Nicole71	
	38	Jordyn.Jacobson2	

2. **Inactive User Engagement:** The team wants to encourage inactive users to start posting by sending them promotional emails.

To identify users who have never posted a single photo on Instagram:

```
SELECT * FROM users
WHERE id NOT IN
(
SELECT user_id FROM photos
);
```

Result Grid				Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id	username	created_at				
▶	5	Aniya_Hackett	2016-12-07 01:04:39				
	7	Kassandra_Homenick	2016-12-12 06:50:08				
	14	Jadyn81	2017-02-06 23:29:16				
	21	Rocio33	2017-01-23 11:51:15				
	24	Maxwell.Halvorson	2017-04-18 02:32:44				
	25	Tierra.Trantow	2016-10-03 12:49:21				
	34	Pearl7	2016-07-08 21:42:01				
	36	Ollie_Ledner37	2016-08-04 15:42:20				
	41	Mckenna17	2016-07-17 17:25:45				
	45	David.Osinski47	2017-02-05 21:23:37				
	49	Morgan.Kassulke	2016-10-30 12:42:31				
	53	Linnea59	2017-02-07 07:49:34				
	54	Duane60	2016-12-21 04:43:38				
	57	Julien_Schmidt	2017-02-02 23:12:48				
	66	Mike.Auer39	2016-07-01 17:36:15				
	68	Franco_Keebler64	2016-11-13 20:09:27				
	71	Nia_Haag	2016-05-14 15:38:50				
	74	Hulda.Macejkovic	2017-01-25 17:17:28				
	75	Leslie67	2016-09-21 05:14:01				
	76	Janelle.Nikolaus81	2016-07-21 09:26:09				
	80	Darby_Herzog	2016-05-06 00:14:21				
	81	Esther.Zulauf61	2017-01-14 17:02:34				
	83	Bartholome.Bernhard	2016-11-06 02:31:23				
	89	Jessyca_West	2016-09-14 23:47:05				
	90	Esmeralda.Mraz57	2017-03-03 11:52:27				
	91	Bethany20	2016-06-03 23:31:53				
*	NULL	NULL	NULL				

3. **Contest Winner Declaration:** The team has organized a contest where the user with the most likes on a single photo win.

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

```
SELECT photo_id FROM likes
GROUP BY photo_id
ORDER BY COUNT(user_id) DESC
LIMIT 1;
```

To retrieve the winner details

```
WITH MostLikedPhotos as
(
SELECT photo_id, COUNT(user_id) AS total_likes FROM likes
GROUP BY photo_id
ORDER BY COUNT(user_id) DESC
LIMIT 1
)
SELECT u.username, u.id, p.id as photo_id, MostLikedPhotos.total_likes from MostLikedPhotos
JOIN photos p on MostLikedPhotos.photo_id= p.id
JOIN users u on u.id= p.user_id;
```

In the first part, we are querying the **likes** table to identify the photo that has received the most likes. The **SELECT** statement retrieves the **photo\_id** from the **likes** table. The **GROUP BY** clause groups the data by **photo\_id**. The **ORDER BY** clause arranges the grouped results in descending order of the count of **user\_id** (which represents the likes). So, the most liked photo will be at the top. The **LIMIT 1** clause ensures that you only get the top result, which is the photo with the most likes.

In the second part, we are building upon the previous query to retrieve the details of the user who posted the most liked photo. The **WITH** clause creates a common table expression (CTE) named **MostLikedPhoto**. This CTE uses the same logic as the first query to find the most liked photo but also calculates the total number of likes for that photo and stores it in the **total\_likes** column. Now we're joining the **MostLikedPhoto** CTE with the **photos** and **users** tables. This allows us to fetch details like the username (**u.username**), user ID (**u.id**), photo ID (**p.id**), and the total number of likes (**MostLikedPhoto.total\_likes**).

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
username	id	photo_id	total_likes
Zack_Kemmer93	52	145	48



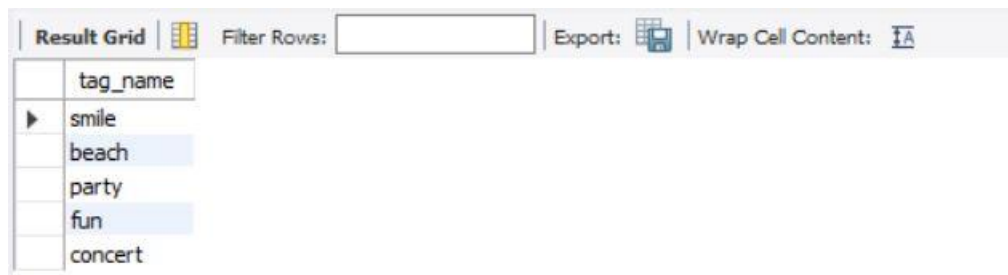
4. **Hashtag Research:** A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

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

```
SELECT * FROM tags;
WITH top_tags AS
(
SELECT tag_id FROM photo_tags
GROUP BY tag_id
ORDER BY COUNT(tag_id) DESC
LIMIT 5
)
SELECT t.tag_name FROM top_tags
JOIN tags t on t.id= top_tags.tag_id;
```

Here we are using **WITH** clause creating a Common Table Expression (CTE) named **top\_tags** to identify the top five most commonly used hashtags. From **photo\_tags** table, the **GROUP BY** clause groups the data by **tag\_id** counting how many times each tag has been used. The **ORDER BY** clause arranges the groups in descending order of the count of **tag\_id** occurrences, which means the most commonly used tags will be at the top. The **LIMIT 5** clause ensures you only retrieve the top five tags.

In the final part of the query, we're joining the **top\_tags** CTE with the **tags** table to retrieve the names of the top tags. This allows us to present the actual hashtag names that are the most commonly used on the platform.



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar, an 'Export' button, and a 'Wrap Cell Content' toggle. The result grid displays a table with two columns: 'tag\_name' and an unlabeled column. The data rows are: smile, beach, party, fun, and concert. The 'smile' and 'fun' rows are highlighted with a blue background.

tag_name	
smile	
beach	
party	
fun	
concert	

5. **Ad Campaign Launch:** The team wants to know the best day of the week to launch ads.

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

```
SELECT dayname(created_at) as days_of_week, COUNT(*) AS no_of_users_registers FROM users
GROUP BY dayname(created_at)
ORDER BY no_of_users_registers DESC;
```

In the above query, to determine the date of the week, we use **DAYNAME** function which is applied to the **created\_at** column as it's a timestamp column. The **count(\*)** function calculates the number of users registered on each day of the week. The **GROUP BY** clause groups the

data by the day of the week to aggregate the registration counts. The **ORDER BY** clause arranges the results in descending order based on the count of registered users.

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
days_of_week	no_of_users_registers		
Thursday	16		
Sunday	16		
Friday	15		
Tuesday	14		
Monday	14		
Wednesday	13		
Saturday	12		

The insights gained from this query will help us determine the best day of the week to launch ad campaigns, as we'll be able to identify the day when most users tend to register on Instagram. This information can guide your ad campaign scheduling strategy for maximum impact.

## B) Investor Metrics:

1. **User Engagement:** Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

To calculate the average number of posts per user on Instagram and to provide the total number of photos on Instagram divided by the total number of users.

```
SELECT COUNT(id) AS total_photos FROM photos; #Total number of photos
SELECT COUNT(DISTINCT id) AS total_users FROM users; #Total number of users
SELECT COUNT(id) / COUNT(DISTINCT user_id) AS avg_posts_per_user
FROM photos; #Photos per user
```

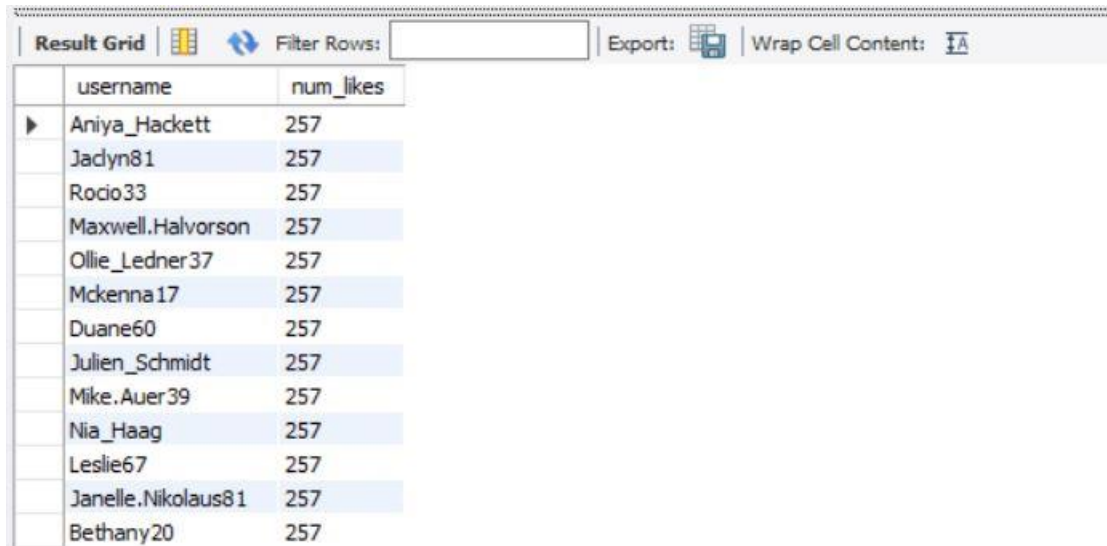
Result Grid	Filter Rows:	Export:	Wrap Cell Content:
avg_posts_per_user			
3.4730			

2. **Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts.

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

```
select username, count(*) as num_likes
from users u
join likes l
on u.id = l.user_id
group by l.user_id
having num_likes = (select count(*) from photos)
```

It joins the users and likes table, then groups the data by **user\_id**, and filters the results to only include users who have liked the same number of photos as the total number of photos available in the photos table.



	username	num_likes
▶	Aniya_Hackett	257
	Jadlyn81	257
	Rocio33	257
	Maxwell.Halvorson	257
	Ollie_Ledner37	257
	Mckenna17	257
	Duane60	257
	Julien_Schmidt	257
	Mike.Auer39	257
	Nia_Haag	257
	Leslie67	257
	Janelle.Nikolaus81	257
	Bethany20	257

## **RESULTS:**

I gained a lot of valuable insights from this project as it gave a glimpse of how business projects work and how a company derive its data from SQL queries and implement them in real time which results in increasing their overall productivity. The results of the analysis will be helpful to various departments.

This Instagram user analytics project provided insights on marketing, user engagements, investor metrics. These insights can be used by Instagram product team to launch new campaigns, track users' engagement and improve user experience.

The insights I derived in these projects:

1. Top 5 Oldest Users of Instagram
2. Users who never posted photos on Instagram
3. Most liked photo on Instagram
4. Top 5 most commonly used hash-tags on Instagram
5. Total number of users on Instagram
6. Total number of photos on Instagram
7. Average number of photos per user
8. Bots and fake accounts on Instagram