



# ***Instagram User Analytics***

**BY  
SHNGAIN KUPAR SHULLAI**

## ***Overview of the Project***

- ✓ People use Instagram in various ways that contribute to making the platform better.
- ✓ I look at things like how often people use the app and what they do while using it.
- ✓ This information can be used by different teams at Instagram, like marketing, product, and development, to make decisions.
- ✓ Instagram users play a crucial role in shaping the Instagram experience through their content creation, engagement, feedback, advocacy, collaboration, and reporting efforts.
- ✓ Their active participation contributes to making Instagram a dynamic, diverse, and inclusive platform for self-expression, connection, and discovery.
- ✓ For example, it can help decide what new features to add or how to make the app better for everyone.

## ***Tech-Stack Used***

- ✓ I use MS Word 2021, Google Drive, MySQL Workbench 8.0 CE to do this analysis.
- ✓ My goal is to find useful information from the data that can help improve Instagram in the future.
- ✓ SQL is like a common language for computers, making it easy to use and learn.

## ***Why I chose SQL for this case study?***

- ✓ With SQL, I can quickly find exactly what we're looking for in Instagram data.
- ✓ Instagram has tons of data, but SQL it can handle it.
- ✓ SQL helps me find how different parts of Instagram data are related, showing us interesting patterns.
- ✓ SQL databases have built-in ways to keep Instagram data safe from unauthorized access.
- ✓ I am deeply passionate about honing my skills with SQL Workbench and other related software tools.
- ✓ My goal is to steadily progress and refine my abilities until I reach a professional level of proficiency.

# ***1. Marketing Analysis***

## ***A. Loyal User Reward***

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

### ***My Task***

Identify the five oldest users on Instagram from the provided database.

### ***Approach***

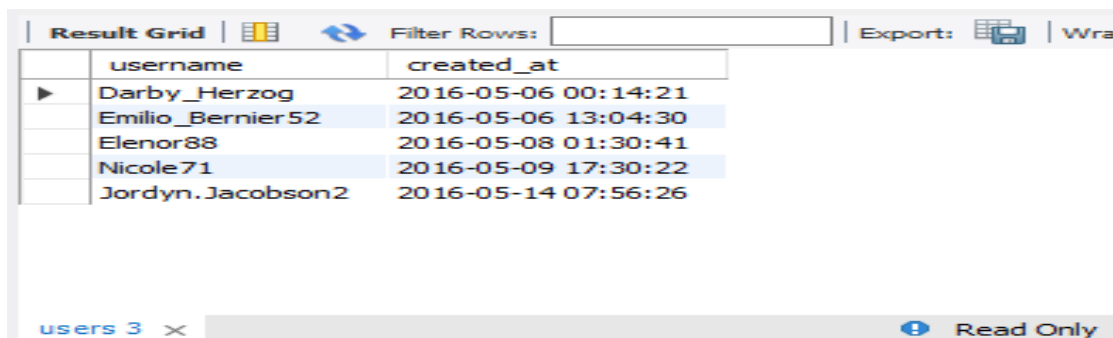
***\*\*\*Screenshots Attached***

- ❖ Now I need to identify 5 oldest Insta users from the database.
- ❖ So, I need to write a SQL Query to select the earliest signups dates.
- ❖ I have a table named 'users' in my database containing signups information.

Therefore, I am applying the following query:

```
23 • SELECT
24     username,
25     created_at
26 FROM
27     users
28 ORDER BY
29     created_at
30 LIMIT 5
31 ;
32
```

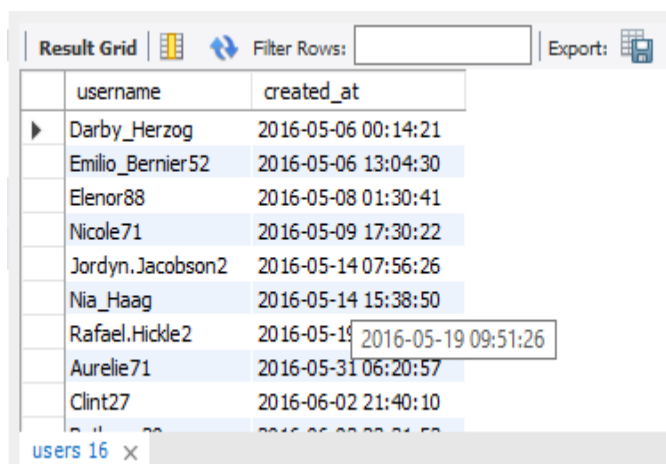
This query selects the stated columns from the stated table, order the results by 'created\_at' column in the ascending order so that the earliest signups can appear first and then it also limits the order results by giving us the earliest first five rows of signups.



The screenshot shows a database interface with a 'Result Grid' tab. It displays the first 5 rows of the 'users' table, ordered by the 'created\_at' column in ascending order. The columns are 'username' and 'created\_at'. The data is as follows:

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

At the bottom, it says 'users 3' and 'Read Only'.



The screenshot shows a database interface with a 'Result Grid' tab. It displays the first 16 rows of the 'users' table, ordered by the 'created\_at' column in ascending order. The columns are 'username' and 'created\_at'. The data is as follows:

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
Nia_Haag	2016-05-14 15:38:50
Rafael.Hickle2	2016-05-19 09:51:26
Aurelie71	2016-05-31 06:20:57
Clint27	2016-06-02 21:40:10

At the bottom, it says 'users 16'.

Similarly, this query results by 'created\_at' column in the ascending order so that the earliest signups can appear first and then the order results by giving us all the signups starting from the earliest one.

```
32
33 • SELECT username, created_at
34 FROM users order by created_at asc;
35
--
```

Therefore, the first five users are the oldest users.

### ***Insights***

- ❖ From the Insta users' registration patterns, I understand the importance of data analysis when it extracts users' behaviour and decision-making steps and actions.
- ❖ I then identify potential growth chances and targeting to market sections of specific Insta users.

## ***B. Inactive User Engagement***

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

### ***My Task***

Identify users who have never posted a single photo on Instagram.

### ***Approach:***

***\*\*\*Screenshots Attached***

- ❖ Now I need to identify Insta users who have never posted a single photo from the database.
- ❖ So, I need to write a SQL Query to identify users who have no records of photos or have not uploaded any photos in their Instagram account.

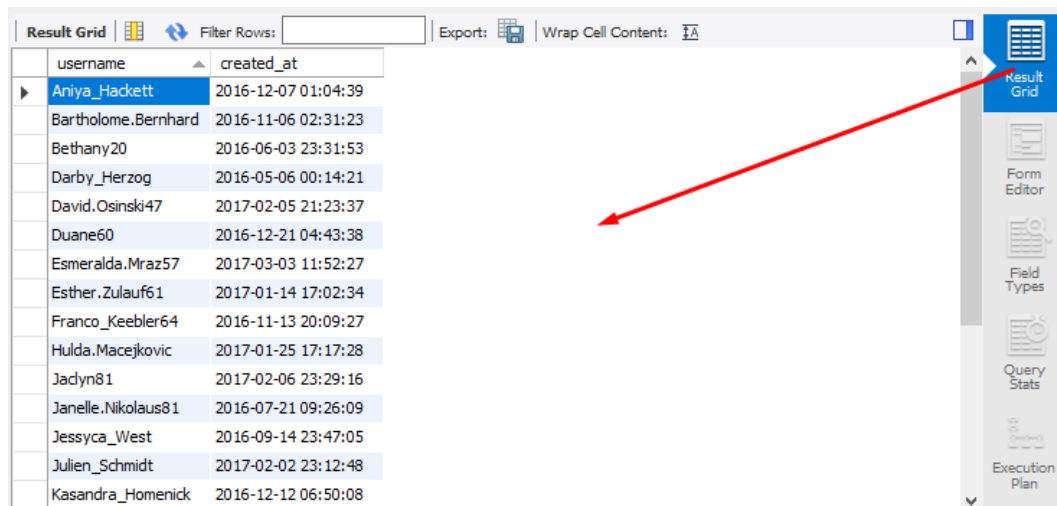
I am applying the following query:

```
128 • SELECT
129     username, created_at
130 FROM
131     users
132     LEFT JOIN
133     photos ON users.id = photos.user_id
134 WHERE
135     photos.id IS NULL;
```

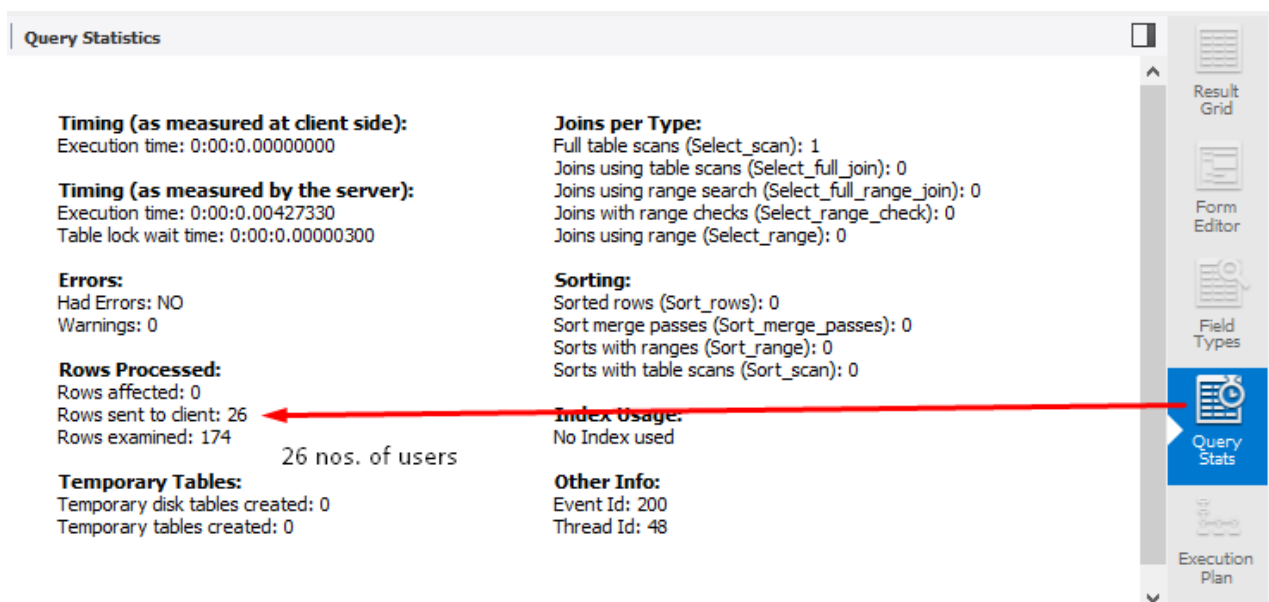
```
SELECT
    username, created_at
FROM
    users
LEFT JOIN
    photos
ON users.id = photos.user_id
WHERE
    photos.id IS NULL;
```

- ❖ This query will filter only those users or rows of 'id' column in the 'photos' table who do not have any entries in the 'photos' table with the help of left join between 'users' and 'photos' tables.
- ❖ 'username' and 'created\_at' are two columns for the results set.
- ❖ 'created\_at' entitled for the signup dates.
- ❖ 'users' table is the main table of data.
- ❖ left join between 'users' and 'photos' tables based on 'id' column in 'users' and 'user\_id' column in 'photos' table.
- ❖ Then 'where' clause filters the results of 'id' column in the 'photos' table which is null or which do not have any entries and thus comprised in the results set.

❖



username	created_at
Aniya_Hackett	2016-12-07 01:04:39
Bartholome.Bernhard	2016-11-06 02:31:23
Bethany20	2016-06-03 23:31:53
Darby_Herzog	2016-05-06 00:14:21
David.Osinski47	2017-02-05 21:23:37
Duane60	2016-12-21 04:43:38
Esmeralda.Mraz57	2017-03-03 11:52:27
Esther.Zulauf61	2017-01-14 17:02:34
Franco_Keebler64	2016-11-13 20:09:27
Hulda.Macejkovic	2017-01-25 17:17:28
Jadyn81	2017-02-06 23:29:16
Janelle.Nikolaus81	2016-07-21 09:26:09
Jessyca_West	2016-09-14 23:47:05
Julien_Schmidt	2017-02-02 23:12:48
Kasandra_Homenick	2016-12-12 06:50:08



Query Statistics	
<b>Timing (as measured at client side):</b> Execution time: 0:00:0.00000000  <b>Timing (as measured by the server):</b> Execution time: 0:00:0.00427330 Table lock wait time: 0:00:0.00000300  <b>Errors:</b> Had Errors: NO Warnings: 0  <b>Rows Processed:</b> Rows affected: 0 Rows sent to client: 26 Rows examined: 174  <b>Temporary Tables:</b> Temporary disk tables created: 0 Temporary tables created: 0	<b>Joins per Type:</b> Full table scans (Select_scan): 1 Joins using table scans (Select_full_join): 0 Joins using range search (Select_full_range_join): 0 Joins with range checks (Select_range_check): 0 Joins using range (Select_range): 0  <b>Sorting:</b> Sorted rows (Sort_rows): 0 Sort merge passes (Sort_merge_passes): 0 Sorts with ranges (Sort_range): 0 Sorts with table scans (Sort_scan): 0  <b>Index Usage:</b> No Index used  <b>Other Info:</b> Event Id: 200 Thread Id: 48



## ***Insights***

- ❖ There are 26 numbers of Insta users who have never posted a single photo on Instagram.
- ❖ There is a section of Insta users who do not give any attention in content creation.
- ❖ This section of users' needs encouragement in the form of personalized approach strategies of creation based and overall users' journey in order to increase their participation.
- ❖ I therefore advice much efforts of the Application Development Team to build or create features to motivate less attentive users.
- ❖ Overall, I discovered that users are highly engaged, and there are many ways to get specific information from the database, leading to valuable insights about content effectiveness and user preferences from popular posts.

## C. Contest Winner Declaration

The team has organized a contest where the user with the most likes on a single photo win.

### My Task

Determine the winner of the contest and provide their details to the team.

### Approach:

**\*\*\*Screenshots Attached**

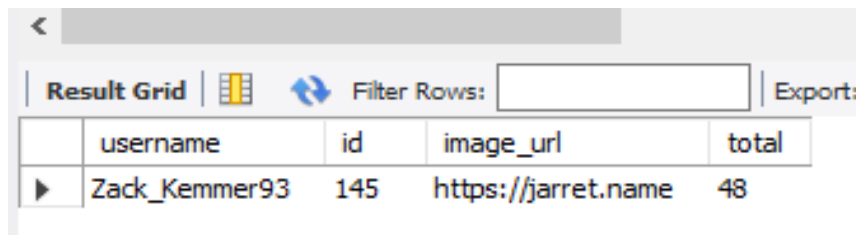
- ❖ Now I need to Determine the winner of the contest and provide the details of the winner to the team from the database.
- ❖ So, I need to write a SQL Query to identify one user who have the highest number of likes from any of his posted photo in their Instagram account.

I am applying the following query:

```
43
44 • SELECT
45     users.username,
46     photos.id,
47     photos.image_url,
48     COUNT(likes.user_id) AS total
49 FROM
50     photos
51     JOIN
52     likes ON likes.photo_id = photos.id
53     JOIN
54     users ON photos.user_id = users.id
55 GROUP BY photos.id
56 ORDER BY total DESC
57 LIMIT 1;
58
```

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

- ❖ This query will order the results of the total number of likes in descending order but limiting the results set to only one row.
- ❖ So, only the top-ranked photo or the one with the most likes is returned.
- ❖ 'users.username' : Here 'username' is the column that stores usernames of Insta users. This column is within the 'users' table. It selects the username of the user.



The screenshot shows a database interface with a 'Result Grid' tab. It displays a single row of data with the following columns: username, id, image\_url, and total. The data in the row is: Zack\_Kemmer93, 145, https://jarret.name, and 48.

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

- ❖ 'photos.id' : Here 'id' is the column that stores identity of a photo. This column is within the 'photos' table. It selects the identity of the photo.
- ❖ 'photos.image\_url' : Here 'image\_url' is the column that stores links of photos. This column is within the 'photos' table. It selects the link or url of the photo.
- ❖ Similarly, count(likes.user\_id) : Here 'user\_id' is the column that stores identity of users who liked photos. This column is within the 'likes' table. 'Count()' is a calculation of total number of likes for each photo.
- ❖ Data Join is the combination of information from three tables. i.e., 'photos', 'likes' and 'users'. It matches photos with their likes and connects them with the photos of the users.
- ❖ Group by of 'Photo.id' is considered separately as the identity of each photo altogether.
- ❖ 'ORDER BY total DESC' : 'total' is an alias given to the 'count()' total numbers of likes for each photo which then sorts in descending order.
- ❖ 'LIMIT 1' limits the order results by giving us only one row and making sure I get the photo with the highest number of likes and the user who posted it.
- ❖ The Winner of the contest is: Zack\_Kemmer93  
ID number: 145  
Photo Link: <https://jarret.name>  
Number of likes: 48

## ***Insights***

- ❖ I learned that there's a significant level of engagement from the users.
- ❖ I understand that there are various possibilities in data retrieval system to obtain specific information from the database.
- ❖ There is a valuable insight regarding content effectiveness obtained through the performance of the high ranked photo as well as the user engagement of the respective user.
- ❖ The accuracy of the resulted data enlightened me about bunches of insights in popular contents and user preferences.

## ***D. Hashtag Research***

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

### ***My Task***

Identify and suggest the top five most commonly used hashtags on the platform.

### ***Approach: Attached***

### ***\*\*\*Screenshots***

- ❖ Now I need to Determine the top five most commonly used hashtags on the platform from the database.
- ❖ So, I need to write a SQL Query to identify top five commonly used hashtags on the platform.

I am applying the following query:

```
59
60 • SELECT
61     tags.tag_name, COUNT(*) AS TT
62 FROM
63     tags
64     JOIN
65     photo_tags ON tags.id = photo_tags.tag_id
66 GROUP BY tags.tag_name
67 ORDER BY TT DESC
68 LIMIT 5;
69
```

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

Result Grid		
	tag_name	TT
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

- ❖ This query will order the results of the total count of each hashtag in descending order but limiting the results set to only five rows of the most popular used hashtags.
- ❖ 'tags.tag\_name' : Here 'tag\_name' is the column that stores name of hashtags of Insta users. This column is within the 'tags' table. It selects the name of hashtags of the user.
- ❖ 'tags.tag\_name, COUNT(\*) AS TT': 'TT' is an alias given to the 'count()' total count of each hashtag.
- ❖ Data Join is the combination of information from two tables. i.e., 'tags' and 'photo\_tags'.
- ❖ 'ON tags.id = photo\_tags.tag\_id' : The Join tables is based on the 'id' column in the 'tags' table and matched the 'tag\_id' column in the 'photo\_tags' table.
- ❖ This join attaches hashtags with their related photos.
- ❖ Group by of 'tags.tag\_name' is making sure that each unique hashtag appears once by tag name in the return.
- ❖ 'ORDER BY TT DESC LIMIT 5' : 'TT' is an alias given to the 'count()' total numbers of each hashtag which then sorts in descending order and limits the order results by giving us only five rows of the most popular used hashtags.
- ✓ The top 5 most popular used hashtags that I have identified and suggested in this case study are  
(1)smile, (2) beach (3) party (4) fun and (5) concert.

## ***Insights***

- ❖ I learned that there's a significant level of understanding of the association and structure of data inside the tables.
- ❖ Now, I can slowly execute SQL query skills in many varied or distinct things for data retrieval and analysis.
- ❖ This project showed how looking at data helps us make smart choices.
- ❖ By finding important information in the data, companies can make decisions that make people more interested, help the company grow, and make the app better for everyone who uses it.
- ❖ Discovering the photos that receive the most likes and the users who share them helps me learn what type of content is most popular among Instagram users.
- ❖ Learning about popular hashtags and finding the best-performing users gives me useful information for deciding what to post.
- ❖ By knowing what people like, Instagram users and businesses can make their posts better to get more likes and reach more people.
- ❖ Looking at how people use Instagram and what they like to see shows us bigger patterns in the Instagram community.
- ❖ Knowing these patterns helps us make smart choices about how to improve Instagram, make better ads, and make the app better for everyone.
- ❖ In general, this project showed how important it is to use data to make decisions for a big social media platform. By using SQL skills to look at Instagram data, we can find important information that helps us make good choices and make the platform better for everyone.

## ***E. Ad Campaign Launch***

The team wants to know the best day of the week to launch ads.

### ***My Task***

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

### ***Approach:***

***\*\*\*Screenshots Attached***

- ❖ I need to write a SQL Query to determine the day of the week when most users register on Instagram.

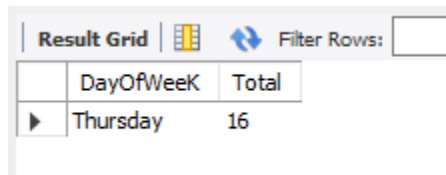
I am applying the following query:

```
72 • SELECT
73     DAYNAME(created_at) AS DayOfWeek, COUNT(*) AS Total
74 FROM
75     users
76 GROUP BY DayOfWeek
77 ORDER BY total DESC
78 LIMIT 1;
```

```
SELECT
    DAYNAME(created_at)
AS      DayOfWeek,
COUNT(*) AS Total
FROM    users
GROUP BY DayOfWeek
ORDER BY total DESC
LIMIT 1;
```



- ❖ This query will sort the days of the week by the number of sign-ups, showing the busiest day first.
- ❖ Then, it only shows the busiest day.



	DayOfWeek	Total
▶	Thursday	16

❖ **DAYNAME(created\_at)** : This section of the query extracts the name of the day from the 'created\_at' column in the 'users' table at the ongoing time of the data records.

- ❖ **AS DayOfWeek** : This part of the query aliases the extracted day name as "DayOfWeek".
  - ❖ **COUNT(\*) AS Total** : This part of the query calculates the total count of users' signups on each day and aliases it as "Total".
  - ❖ **'from users'** : 'users' table
  - ❖ **GROUP BY DayOfWeek** : This part of the query arranges the data so that each day of the week extracted from the 'created\_at' column at the ongoing time of the data records is assembled together.
  - ❖ **ORDER BY total DESC LIMIT 1** : This part of the query sorts the days of the week by the number of user signups, showing the busiest day first. Then, it only limits the busiest day.
- ✓ So, Thursday is the day when 16 users signed up on Instagram.

## ***Insights***

- ❖ If Thursday has the most sign-ups, it might mean more people are active on that day.
- ❖ So, running ad campaigns on Thursdays could reach more people and be more effective, since more users are active on the platform during that time.
- ❖ Moreover, taking into account the characteristics and actions of the specific audience I'm targeting can help tailor ad campaign schedules to match the times when they're most likely to be engaged.
- ❖ Analysing the trends in user sign-ups on Instagram offers valuable guidance for scheduling ad campaigns effectively.
- ❖ Recognizing the days when user engagement is highest allows businesses to strategically time their advertising efforts for maximum reach and impact.
- ❖ Moreover, tailoring ad campaign schedules to match peak engagement periods based on the demographics and behaviours of the target audience can further optimize campaign performance and overall success.

## 2) Investor Metrics

### A. User Engagement

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

**Task 1** Calculate the average number of posts per user on Instagram.

**Approach:**

**\*\*\*Screenshots Attached**

- ❖ I need to write a SQL Query to determine the average number of posts per distinct users to avoid (multiple times) same users' posts who have posted photos on Instagram.

I am applying the following query:

```
10
11 • SELECT
12     COUNT(*) / (SELECT COUNT(DISTINCT user_id)
13     FROM photos) AS average_posts_per_user
14 FROM
15     photos;
16
```

```
SELECT

    COUNT(*) / (SELECT COUNT(DISTINCT user_id)

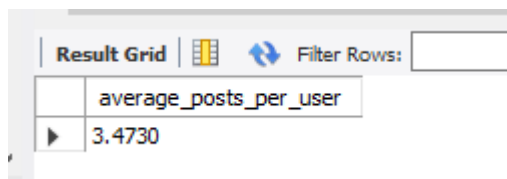
FROM photos)

AS    average_posts_per_user

FROM

    photos;
```

- ❖ This SQL query finds the average number of posts per user on Instagram.
- ❖ It does this by dividing the total number of posts by the count of different users who have posted photos.
- ❖ (SELECT COUNT(DISTINCT user\_id) FROM photos) : This inner part of the query counts how many different users have posted photos on Instagram by looking at their unique user ids.
- ❖ COUNT(\*) : This outer part of the query counts the total number of posts made by all users on Instagram.
- ❖ COUNT(\*) / (SELECT COUNT(DISTINCT user\_id) FROM photos) : The number from the outer part of the query is divided by the number from the inner part to figure out how many posts each user makes on average.
- ❖ AS average\_posts\_per\_user FROM photos : This part of the query aliased the results as 'average\_posts\_per\_user'



The screenshot shows a database interface with a 'Result Grid' tab. It contains a single row with the column name 'average\_posts\_per\_user' and the value '3.4730'.

average_posts_per_user
3.4730

- ✓ Calculate the average number of posts per user on Instagram.
- ✓ On average, each user on Instagram makes 3.4530 posts.

## Task 2

Provide the total number of photos on Instagram divided by the total number of users.

### Approach:

\*\*\*Screenshots Attached

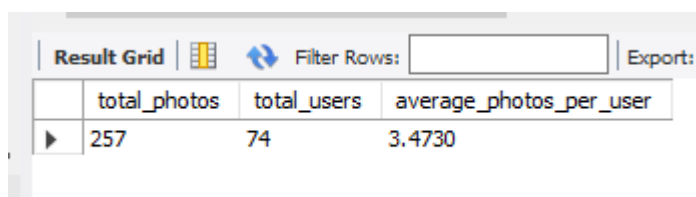
- ❖ I need to write a SQL Query to determine the total number of photos on Instagram divided by the total number of users.

I am applying the following query:

```
18 • SELECT
19     COUNT(*) AS total_photos,
20     COUNT(DISTINCT user_id) AS total_users,
21     COUNT(*) / COUNT(DISTINCT user_id) AS average_photos_per_user
22 FROM
23     photos;
```

```
SELECT COUNT(*) AS total_photos,
        COUNT(DISTINCT user_id) AS total_users,
        COUNT(*) / COUNT(DISTINCT user_id) AS average_photos_per_user
FROM    photos;
```

- ❖ COUNT(\*) AS total\_photos : The COUNT(\*) function counts all the photos that are present and aliased as 'total\_photos'
- ❖ COUNT(DISTINCT user\_id) AS total\_users : This function counts how many distinct users have posted photos and aliased as 'total\_users'
- ❖ COUNT(\*) / COUNT(DISTINCT user\_id) AS average\_photos\_per\_user FROM photos : This part finds out how many photos each user has posted on average by dividing the total number of photos by the total number of users. It aliased as 'average\_photos\_per\_user'



	total_photos	total_users	average_photos_per_user
▶	257	74	3.4730

- ✓ Provide the total number of photos on Instagram divided by the total number of users.
- ✓  $257 / 74 = 3.473$

## ***B. Bots & Fake Accounts:***

Investors want to know if the platform is crowded with fake and dummy 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.

### ***Approach:***

***\*\*\*Screenshots Attached***

- ❖ I need to write a SQL Query to determine the average number of posts per distinct users to avoid (multiple times) same users' posts who have posted photos on Instagram.

Here's a SQL query to do that:

```
38 FROM (  
39     SELECT user_id, COUNT(*) AS total_likes  
40     FROM likes  
41     GROUP BY user_id  
42 ) AS user_likes  
43 JOIN (  
44     SELECT COUNT(*) AS total_photos  
45     FROM photos  
46 ) AS total_photos  
47 ON user_likes.total_likes = total_photos.total_photos;
```

```
SELECT      user_id  
  
FROM (SELECT user_id, COUNT(*) AS total_likes  
  
FROM      likes  
  
GROUP BY user_id) AS user_likes  
  
JOIN (SELECT COUNT(*) AS total_photos  
  
FROM photos) AS total_photos  
  
ON user_likes.total_likes = total_photos.total_photos;
```

Result Grid	
	user_id
▶	5
	14
	21
	24
	36
	41
	54
	57
	66
	71
	75
	76
	91

- ❖ To spot potential bots or fake accounts on Instagram, we'll look for users who have liked every single photo on the site.
- ❖ This behaviour isn't common for regular users.
- ❖ FROM (SELECT user\_id, COUNT(\*) AS total\_likes FROM likes GROUP BY user\_id) AS user\_likes : This inner query performs a calculation to determine the cumulative likes attributed to each user.
- ❖ By counting the occurrences of likes associated with each user, it aliased as "total\_likes" for each user.
- ❖ It achieves this by organizing the data based on the user\_id column within the likes table, effectively grouping the likes data for each user.
- ❖ This result is aliased as "user\_likes", facilitating further operations on the aggregated like counts per user.
- ❖ This section of the query calculates the total number of photos in the photos table. It does this by counting all the rows in the photos table and assigning the result the alias "total\_photos".
- ❖ JOIN ( SELECT COUNT(\*) AS total\_photos FROM photos) AS total\_photos : This outer query calculates the total number of photos in the photos table. It does this by counting all the rows in the photos table and aliased as "total\_photos".
- ❖ ON user\_likes.total\_likes = total\_photos.total\_photos : In this main query, the results of the two subqueries are combined based on a condition: the total number of likes for a user must match the total number of photos. This condition ensures that only users who have liked every single photo are included in the final result.
- ✓ The query outputs the user IDs of users who have liked an equivalent number of photos as the total count of photos present on the site. This means that these users have liked every single photo on the platform, as their total number of likes matches the total number of photos available.
- ✓ These 13 accounts are potentially fake or spam accounts that are overwhelming the application.

## ***Insights***

### ***Average Number of Posts per User:***

- ❖ This metric offers valuable insights into user engagement and activity levels observed on the platform.
- ❖ A higher average signifies the presence of active users who consistently participate in posting content on Instagram.
- ❖ On the contrary, a lower average could indicate reduced user engagement or a smaller community of users on the platform.

### ***Total Number of Photos per User***

- ❖ A higher ratio suggests that users are more active in posting photos, which could indicate influencers or highly engaged content creators within the community.
- ❖ In contrast, a lower ratio might imply users who are less active in sharing photos, potentially affecting the overall level of engagement across the platform.

### ***Identifying Fake and Dummy Accounts:***

- ❖ Analyzing accounts displaying suspicious activities is crucial for mitigating the potential risks associated with fake engagement. This approach ensures the preservation of the platform's authenticity and integrity.
- ❖ Identifying unusual patterns of behavior, like excessive liking or posting, can serve as an indicator of bots or fraudulent accounts on the platform.
- ❖ Reviewing user behavior, such as consistently liking every photo, can assist in identifying potential fake or spam accounts.



## ***Results***

- ❖ Through the "Instagram User Analytics" project, we gained valuable insights into how users engage with the platform.
- ❖ By looking at things like how often users post, what content they engage with, and any unusual behaviour, we were able to learn a lot about user habits.
- ❖ These insights helped us make smarter decisions.
- ❖ For example, we could better plan ad campaigns by scheduling them on days when users are most active. We also identified and dealt with potential fake accounts or bots, ensuring the platform remains authentic and trustworthy.
- ❖ Understanding user preferences and behaviours allowed us to improve the overall user experience. By tailoring features and content to match what users want, we could enhance satisfaction and engagement.

## ***Conclusion***

- ✓ Overall, this project showed us how powerful data analysis can be in shaping the success of a social media platform like Instagram.
- ✓ By using data to guide our decisions, we can make the platform better for everyone involved.

