**Objective Questions & Answers**

1. Are there any tables with duplicate or missing null values? If so, how would you handle them?

**Code**: -- Check for duplicate User IDs (should return 0 rows)

SELECT id, COUNT(\*)

FROM users

GROUP BY id

HAVING COUNT(\*) > 1;

-- Check for duplicate Usernames (should return 0 rows)

SELECT username, COUNT(\*)

FROM users

GROUP BY username

HAVING COUNT(\*) > 1;

-- Check for duplicate Photo IDs (should return 0 rows)

SELECT id, COUNT(\*)

FROM photos

GROUP BY id

HAVING COUNT(\*) > 1;

-- Check for duplicate Comment IDs (should return 0 rows)

SELECT id, COUNT(\*)

FROM comments

GROUP BY id

HAVING COUNT(\*) > 1;

-- Check for duplicate Tag IDs (should return 0 rows)

SELECT id, COUNT(\*)

FROM tags

GROUP BY id

HAVING COUNT(\*) > 1;

-- Check for duplicate Tag Names (should return 0 rows)

SELECT tag\_name, COUNT(\*)

FROM tags

GROUP BY tag\_name

HAVING COUNT(\*) > 1;

-- Check for duplicate Likes (user\_id, photo\_id) - Schema PK should prevent this (should return 0 rows)

SELECT user\_id, photo\_id, COUNT(\*)

FROM likes

GROUP BY user\_id, photo\_id

HAVING COUNT(\*) > 1;

-- Check for duplicate Follows (follower\_id, followee\_id) - Schema PK should prevent this (should return 0 rows)

SELECT follower\_id, followee\_id, COUNT(\*)

FROM follows

GROUP BY follower\_id, followee\_id

HAVING COUNT(\*) > 1;

-- Check for duplicate Photo Tags (photo\_id, tag\_id) - Schema PK should prevent this (should return 0 rows)

SELECT photo\_id, tag\_id, COUNT(\*)

FROM photo\_tags

GROUP BY photo\_id, tag\_id

HAVING COUNT(\*) > 1;

-- ------------- NULL VALUE CHECKS (for NOT NULL columns) -------------

-- Check for NULLs in users table (should return 0)

SELECT COUNT(\*) AS null\_count FROM users WHERE id IS NULL OR username IS NULL;

-- Check for NULLs in photos table (should return 0)

SELECT COUNT(\*) AS null\_count FROM photos WHERE id IS NULL OR image\_url IS NULL OR user\_id IS NULL;

-- Check for NULLs in comments table (should return 0)

SELECT COUNT(\*) AS null\_count FROM comments WHERE id IS NULL OR comment\_text IS NULL OR user\_id IS NULL OR photo\_id IS NULL;

-- Check for NULLs in likes table (PK columns cannot be NULL - check is redundant but safe) (should return 0)

SELECT COUNT(\*) AS null\_count FROM likes WHERE user\_id IS NULL OR photo\_id IS NULL;

-- Check for NULLs in follows table (PK columns cannot be NULL - check is redundant but safe) (should return 0)

SELECT COUNT(\*) AS null\_count FROM follows WHERE follower\_id IS NULL OR followee\_id IS NULL;

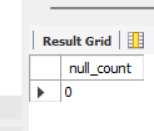
-- Check for NULLs in tags table (should return 0)

SELECT COUNT(\*) AS null\_count FROM tags WHERE id IS NULL OR tag\_name IS NULL;

-- Check for NULLs in photo\_tags table (PK columns cannot be NULL - check is redundant but safe) (should return 0)

SELECT COUNT(\*) AS null\_count FROM photo\_tags WHERE photo\_id IS NULL OR tag\_id IS NULL;

**SQL Output:**



**Answer Explanation**: No, there is no table having duplicate or missing null values in our dataset. But, if there might be any then:

* Duplicates: If duplicates were found in primary or unique key columns, it would indicate a data loading or integrity issue. The standard approach is to investigate the source of duplicates, delete the redundant records (keeping one unique instance), and ensure database constraints (PRIMARY KEY, UNIQUE) are strictly enforced to prevent future occurrences.
* Null Values: If NULL values were found in columns defined as NOT NULL, it would signal a violation of data integrity, possibly due to bypassing constraints during import. These rows would need investigation, correction or removal.

1. What is the distribution of user activity levels (e.g., number of posts, likes, comments) across the user base?

**Code**:

WITH UserActivityCounts AS (

SELECT

u.id AS user\_id,

COUNT(DISTINCT p.id) AS num\_photos\_posted,

COUNT(DISTINCT c.id) AS num\_comments\_made,

COUNT(DISTINCT l.photo\_id) AS num\_likes\_made

FROM

users u

LEFT JOIN

photos p ON u.id = p.user\_id

LEFT JOIN

comments c ON u.id = c.user\_id

LEFT JOIN

likes l ON u.id = l.user\_id

GROUP BY

u.id

)

SELECT

'Photo Posting' AS activity\_type,

MIN(num\_photos\_posted) AS min\_count,

MAX(num\_photos\_posted) AS max\_count,

AVG(num\_photos\_posted) AS avg\_count,

SUM(CASE WHEN num\_photos\_posted = 0 THEN 1 ELSE 0 END) AS zero\_activity\_users,

COUNT(\*) AS total\_users -- Total users

FROM UserActivityCounts

UNION ALL

SELECT

'Commenting' AS activity\_type,

MIN(num\_comments\_made) AS min\_count,

MAX(num\_comments\_made) AS max\_count,

AVG(num\_comments\_made) AS avg\_count,

SUM(CASE WHEN num\_comments\_made = 0 THEN 1 ELSE 0 END) AS zero\_activity\_users,

COUNT(\*) AS total\_users

FROM UserActivityCounts

UNION ALL

SELECT

'Liking' AS activity\_type,

MIN(num\_likes\_made) AS min\_count,

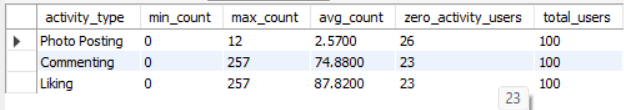
MAX(num\_likes\_made) AS max\_count,

AVG(num\_likes\_made) AS avg\_count,

SUM(CASE WHEN num\_likes\_made = 0 THEN 1 ELSE 0 END) AS zero\_activity\_users,

COUNT(\*) AS total\_users

FROM UserActivityCounts;

**SQL Output**: 

**Answer Explanation**:

Distribution of User Activity Levels

To understand the distribution of user activity, we calculated the number of photos posted, comments made, and likes given for each user in the database. This was achieved using the above SQL query.

* Photo Posting:
  + 1. The maximum & minimum number of photos posted by a user are 12 & 0 respectively.
    2. The average number of photos posted per user is approximately 2.57.
    3. A significant portion of users 26 out of 100 total users have posted zero photos, indicating a potential group of less active content creators or primarily content consumers.
* Commenting:
  + 1. The number of comments made per user ranges from 0 to 257.
    2. The average number of comments made per user is approximately 74.88.
    3. Similar to posting, 23 users have made zero comments.
* Liking:
  + 1. The number of likes given per user ranges from 0 to 257.
    2. The average number of likes given per user is approximately 87.82.
    3. 23 users have not liked any photos.

1. Calculate the average number of tags per post (photo\_tags and photos tables).

**Code**:

WITH TagsPerPhoto AS (

SELECT

p.id AS photo\_id,

-- COUNT(pt.tag\_id) correctly counts 0 for photos with no tags due to LEFT JOIN.

COUNT(pt.tag\_id) AS num\_tags

FROM

photos p

LEFT JOIN

photo\_tags pt ON p.id = pt.photo\_id

GROUP BY

p.id

)

-- Calculate the average

SELECT

ROUND(AVG(num\_tags), 2) AS average\_tags\_per\_post

FROM

TagsPerPhoto;

**SQL Output**:

C:\Users\Lenovo\OneDrive\Pictures\Screenshots\Screenshot 2025-06-22 133354.png

**Answer Explanation**: To determine the average number of tags associated with each post, we calculated the count of tags for every photo and then averaged these counts. This involved a LEFT JOIN from the photos table to the photo\_tags table to include photos with zero tags, grouping by photo ID, and counting the associated tags (COUNT(pt.tag\_id)). Finally, the AVG() function was applied to these counts.

* The average number of tags used per post across the entire dataset is approximately 1.95.

1. Identify the top users with the highest engagement rates (likes, comments) on their posts and rank them.

**Code**:

WITH UserPostEngagement AS (

-- Calculate total likes and comments for EACH photo using subqueries

SELECT

p.user\_id, -- The ID of the user who posted the photo

p.id AS photo\_id, -- Need photo\_id to join later

(SELECT COUNT(\*) FROM likes WHERE photo\_id = p.id) AS likes\_received,

(SELECT COUNT(\*) FROM comments WHERE photo\_id = p.id) AS comments\_received

FROM photos p

),

UserTotalEngagement AS (

SELECT

u.id AS user\_id,

u.username,

COALESCE(SUM(upe.likes\_received), 0) AS total\_likes\_received,

COALESCE(SUM(upe.comments\_received), 0) AS total\_comments\_received,

(COALESCE(SUM(upe.likes\_received), 0) + COALESCE(SUM(upe.comments\_received), 0)) AS total\_engagement\_score

FROM users u

LEFT JOIN UserPostEngagement upe ON u.id = upe.user\_id -- Join users to their engagement stats

GROUP BY u.id, u.username

)

-- Final selection and ranking

SELECT

user\_id,

username,

total\_likes\_received,

total\_comments\_received,

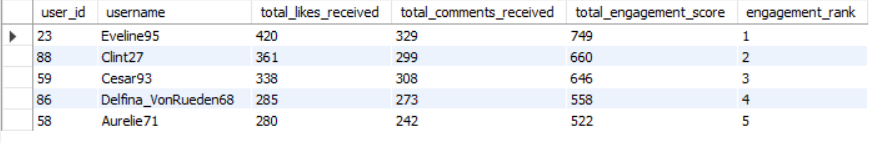
total\_engagement\_score,

RANK() OVER (ORDER BY total\_engagement\_score DESC) AS engagement\_rank

FROM UserTotalEngagement

ORDER BY engagement\_rank ASC, user\_id ASC

LIMIT 5;

**SQL Output**: 

**Answer Explanation:** To identify users receiving the highest total volume of engagement (likes and comments) on their posts, we executed the following analysis:

* For each individual photo, we calculated the total number of likes received using (SELECT COUNT(\*) FROM likes WHERE photo\_id = p.id) and the total number of comments received using (SELECT COUNT(\*) FROM comments WHERE photo\_id = p.id).
* These per-photo counts were then aggregated for each user by joining with the users table and summing the likes\_received and comments\_received across all photos posted by that user.
* A 'total engagement score' was calculated for each user by adding their total\_likes\_received and total\_comments\_received.
* Finally, users were ranked in descending order based on this total\_engagement\_score using the RANK() window function.

The analysis revealed the users whose content generated the highest overall volume of interactions. The top 5 users are the following:

1. Which users have the highest number of followers and followings?

**Code**:

SELECT

u.id AS user\_id,

u.username,

-- Subquery to count followers for user u

(SELECT COUNT(\*) FROM follows WHERE followee\_id = u.id) AS followers\_count,

-- Subquery to count how many user u is following

(SELECT COUNT(\*) FROM follows WHERE follower\_id = u.id) AS following\_count

FROM users u

ORDER BY

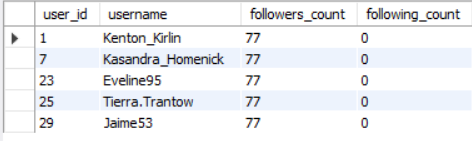
followers\_count DESC,

following\_count DESC,

u.id ASC

LIMIT 5;

**SQL Output**:



**Answer Explanation**: To identify users with the most followers and those who follow the most other users, we queried the users table and used correlated subqueries to count entries in the follows table for each user. Specifically, COUNT(\*) where followee\_id matched the user's ID determined their follower count, and COUNT(\*) where follower\_id matched determined their following count. The results were ordered primarily by follower count in descending order.

* Users with Highest Follower/Following Counts:
* user\_id 1 has the highest number of followers, i.e., 77.

1. Calculate the average engagement rate (likes, comments) per post for each user.

**Code:**

WITH PhotoEngagementScore AS (

-- Calculate total likes and comments for EACH photo

SELECT

p.id AS photo\_id,

p.user\_id,

(SELECT COUNT(\*) FROM likes WHERE photo\_id = p.id) AS likes\_on\_photo,

(SELECT COUNT(\*) FROM comments WHERE photo\_id = p.id) AS comments\_on\_photo

FROM photos p

)

-- Calculate the average engagement score per post for each user

SELECT

u.id AS user\_id,

u.username,

-- Calculate the average of (likes + comments) for all photos posted by user u

COALESCE(AVG(pes.likes\_on\_photo + pes.comments\_on\_photo), 0) AS avg\_engagement\_per\_post

FROM

users u

LEFT JOIN

PhotoEngagementScore pes ON u.id = pes.user\_id

GROUP BY

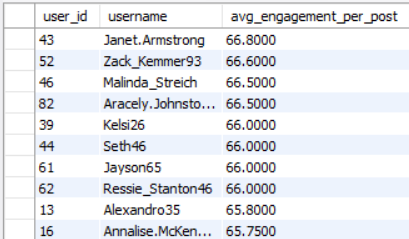
u.id, u.username

ORDER BY

avg\_engagement\_per\_post DESC, -- Show users with highest average engagement first

u.id ASC;

**SQL Output:**



**Answer Explanation:** To understand the typical engagement level generated by individual posts for each user, we calculated the average engagement rate per post. The steps were:

* Determined the total number of likes and comments for each individual photo using subqueries (COUNT(\*)).
* Calculated a total engagement score (likes + comments) for each photo.
* Joined this per-photo data with the users table using a LEFT JOIN to include all users.
* Grouped by user and calculated the AVG()of the per-photo engagement scores across all photos posted by each user. COALESCE was used to assign an average of 0 to users with no posts.

The analysis produced a ranking of users based on the average engagement their posts receive. The users whose posts typically generate the most interaction are:

1. Get the list of users who have never liked any post (users and likes tables)

**Code**:

SELECT

COUNT(\*) AS users\_without\_likes\_count

FROM

users u

WHERE NOT EXISTS (

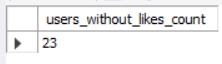
SELECT 1

FROM likes l

WHERE l.user\_id = u.id -- to check if any like exists for this user

);

**SQL Output:**



**Answer Explanation**: To identify users who have not interacted with the platform by liking any posts, we queried for users present in the users table but absent from the likes table. This was achieved using NOT EXISTS. Since, the list might’ve been too long we counted the number of users who have never liked any post and have shown that dataset.

* A total of 23 users were found who have never liked any post.

1. Leveraging User-Generated Content for Ad Campaigns:

**Code**:

-- Q8.a. Strategy: List of Popular Tags

SELECT

tag\_name,

COUNT(\*)

FROM photo\_tags

JOIN tags ON tags.id = photo\_tags.tag\_id

GROUP BY tag\_name

ORDER BY COUNT(\*) DESC;

-- Q8.b. Strategy: Identifying Trending Tags by Recent Likes

-- Define the start date for our 'recent' period.

SET @recent\_cutoff\_date = '2017-04-01'; -- Example cutoff date

SELECT

t.tag\_name,

COUNT(l.user\_id) AS recent\_likes\_count -- Count likes received recently on photos with this tag

FROM

tags t

JOIN

photo\_tags pt ON t.id = pt.tag\_id

JOIN

likes l ON pt.photo\_id = l.photo\_id

WHERE

l.created\_at >= @recent\_cutoff\_date -- Filter likes to only include recent ones

GROUP BY

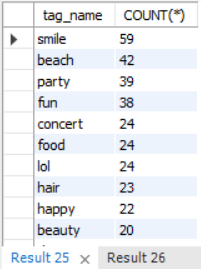
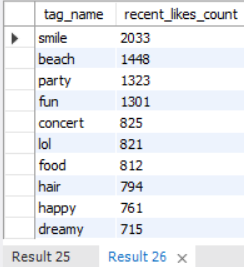
t.tag\_name

ORDER BY

recent\_likes\_count DESC

LIMIT 10;

**SQL Output:**

**Answer Explanation:** The user-generated content within the Instagram dataset, specifically the photos posted, the hashtags used (tags), and their association (photo\_tags), offers significant opportunities for creating more personalized and engaging advertising campaigns. Here's how this data can be leveraged:

* Dynamic Ad Creative using UGC Themes:
* Strategy: Identify trending hashtags or popular user-generated content themes within specific demographics or interest groups.
* Application: Develop ad campaigns that mirror these trending themes or incorporate visuals inspired by popular UGC styles (while respecting user privacy and copyright). For example, if #sunset photos are trending and highly engaged with, an airline might run an ad campaign featuring beautiful sunset destinations.
* We found that “smile”, “beach”, “party” are the top 3 tags receiving the most likes recently which also illustrates the demographics of the type of mood in the instagram algorithm.
* **Conclusion**: By analyzing the explicit interests users declare via hashtags and the implicit interests revealed through their posting behavior, Meta's Marketing team can move beyond broad demographic targeting. Leveraging UGC data allows for highly relevant, personalized ad experiences, which are more likely to resonate with users, increase engagement with ads (clicks, conversions), and improve overall campaign effectiveness and ROI.

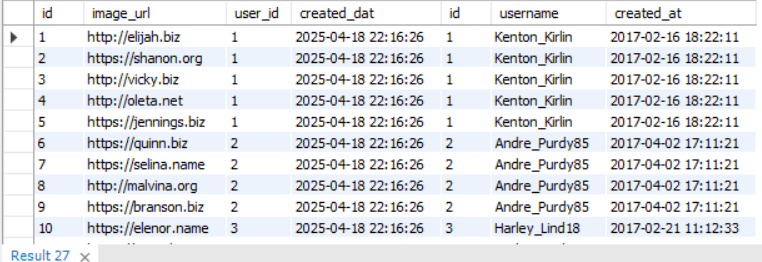
1. Are there any correlations between user activity levels and specific content types (e.g., photos, videos, reels)? How can this information guide content creation and curation strategies?

**Code:**

SELECT \* FROM photos p

JOIN users u ON p.user\_id = u.id; -- does not have a column specifying the content type (e.g., 'photo', 'video', 'reel'). Every entry is essentially treated as a generic "photo" based on the table name and image\_url.

**SQL Output:**



**Answer Explanation:** The question asks for correlations between user activity levels (posting, liking, commenting) and specific content types (photos, videos, reels). However, the provided database schema for the photos table lacks a dedicated column to differentiate between these formats. All content is represented by an image\_url within the photos table***. (Limitation)***

* ***Consequently, a direct analysis correlating user activity with distinct media formats (photo vs. video vs. reel) is not feasible with the current data structure.***
* ***Alternative Analysis (Content Themes via Tags):***
* While we cannot analyze by format, we can explore correlations between user activity and content themes using hashtags as a proxy. For example, we could analyze:
* Do users with higher activity levels (identified in Q2) tend to use or engage with specific categories of tags (e.g., #food, #travel, #fashion) more frequently than less active users?
* Do photos associated with certain popular tags (e.g., the top 10 most used tags) receive significantly different average engagement rates (likes + comments per photo, from Q6)?
* ***Strategic Guidance (Hypothetical Based on Availability of Format Data):***
* Understanding the correlation between user activity and content types/themes is crucial for guiding content creation and curation strategies. If format data were available, or based on the theme analysis via tags, the following strategies could be implemented:
  + 1. Personalized Feed Curation: The platform's recommendation algorithms could prioritize showing users the content formats (or themes/tags) they demonstrably engage with most. If active users heavily engage with video content, their feeds should feature more videos.
    2. Platform Feature Development: If data shows that a specific format (e.g., Reels) drives significantly higher engagement among active users, Meta could prioritize investing more resources into developing features and tools related to that format.

**Conclusion*:*** While the current dataset limits analysis by specific media format, the principle of correlating user activity with content characteristics (be it format or theme) is vital. This understanding allows for more effective personalization, guides platform development, empowers creators, and optimizes marketing efforts by aligning content strategies with user behavior and preferences.

1. Calculate the total number of likes, comments, and photo tags for each user.

**Code**:

WITH PhotoStats AS (

-- Calculate likes, comments, and tags for EACH photo

SELECT

p.id AS photo\_id,

p.user\_id,

(SELECT COUNT(\*) FROM likes WHERE photo\_id = p.id) AS likes\_count,

(SELECT COUNT(\*) FROM comments WHERE photo\_id = p.id) AS comments\_count,

(SELECT COUNT(\*) FROM photo\_tags WHERE photo\_id = p.id) AS tags\_count

FROM photos p

)

SELECT

u.id AS user\_id,

u.username,

COALESCE(SUM(ps.likes\_count), 0) AS total\_likes\_received,

COALESCE(SUM(ps.comments\_count), 0) AS total\_comments\_received,

COALESCE(SUM(ps.tags\_count), 0) AS total\_tags\_used\_on\_posts

FROM

users u

LEFT JOIN -- To Include users even if they have no photos/stats

PhotoStats ps ON u.id = ps.user\_id

GROUP BY

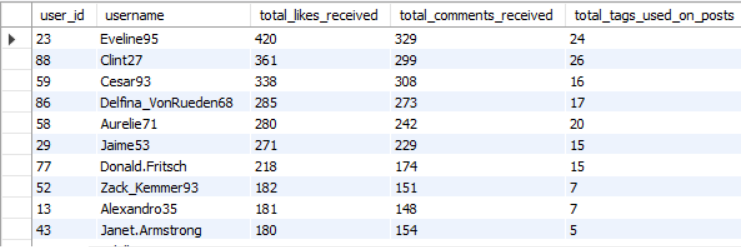
u.id, u.username

ORDER BY

total\_likes\_received DESC,

u.id ASC;

**SQL Output:**

****

**Answer Explanation**: To quantify the total volume of specific interactions associated with each user's posts, we calculated the total likes received, total comments received, and total photo tags used on all photos posted by each user. The methodology involved:

* + 1. Calculating the individual count of likes, comments, and tags for every photo using subqueries within a Common Table Expression (CTE).
    2. Joining this per-photo data with the users table.
    3. Grouping by user and summing the respective counts (likes\_count, comments\_count, tags\_c vount) to get the total for each user. COALESCE was used to handle users with no posts.
* User “Eveline95” received the most likes i.e., 420 as well as the most comments, 329.
* User “Clint27” utilized the highest total number of tags across their posts.

1. Rank users based on their total engagement (likes, comments, shares) over a month.

**Code**:

-- NOTE: "Shares" data is not available in the schema.

-- NOTE: Using April 2025 as the analysis month, based on observed data concentration.

-- Define the start and end dates for April 2025

SET @start\_date = '2025-04-01 00:00:00';

SET @end\_date = '2025-04-30 23:59:59';

WITH MonthlyLikes AS (

-- Count likes MADE BY each user in the specified month

SELECT

user\_id,

COUNT(\*) AS likes\_count

FROM likes

WHERE created\_at BETWEEN @start\_date AND @end\_date

GROUP BY user\_id

),

MonthlyComments AS (

-- Count comments MADE BY each user in the specified month

SELECT

user\_id,

COUNT(\*) AS comments\_count

FROM comments

WHERE created\_at BETWEEN @start\_date AND @end\_date

GROUP BY user\_id

)

-- Combine user data with monthly activity and rank

SELECT

u.id AS user\_id,

u.username,

COALESCE(ml.likes\_count, 0) AS likes\_made\_in\_month,

COALESCE(mc.comments\_count, 0) AS comments\_made\_in\_month,

(COALESCE(ml.likes\_count, 0) + COALESCE(mc.comments\_count, 0)) AS total\_monthly\_engagement,

DENSE\_RANK() OVER (ORDER BY (COALESCE(ml.likes\_count, 0) + COALESCE(mc.comments\_count, 0)) DESC) AS monthly\_engagement\_rank

FROM

users u

LEFT JOIN MonthlyLikes ml ON u.id = ml.user\_id

LEFT JOIN MonthlyComments mc ON u.id = mc.user\_id

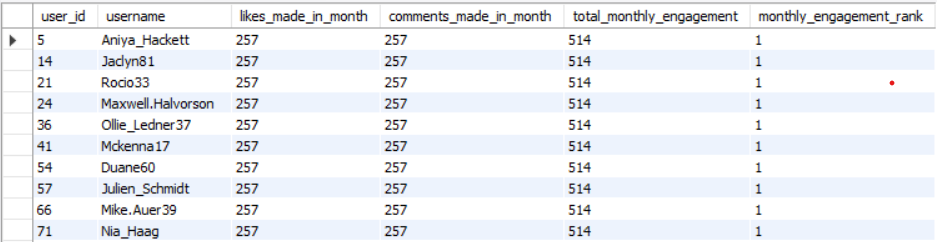
ORDER BY

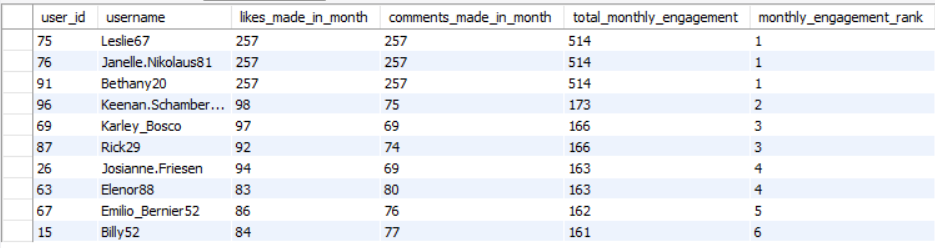
monthly\_engagement\_rank ASC,

u.id ASC

LIMIT 20;

**SQL Output:**

****

****

**Answer Explanation:** To rank users based on their engagement activity within a specific timeframe, we analyzed user actions during April 2025. This month was selected as the primary period of activity observed within the likes and comments data provided in the dataset. Note: ***The requested 'shares' metric could not be included as corresponding data is unavailable in the provided schema. Engagement is therefore defined as the sum of likes given and comments made by the user during April 2025***.

* The methodology involved:
  + 1. Defining the start and end timestamps for April 2025.
    2. It is important to note that all recorded activity (likes and comments) within the provided dataset occurred at a single timestamp ('2025-04-18 22:16:26'). Therefore, while the analysis covers the requested period of April 2025, the engagement reflects activity concentrated entirely at that specific moment.
    3. Using Common Table Expressions (CTEs) to separately count the total likes given and total comments made by each user within this period.
    4. Joining these monthly counts back to the users table using LEFT JOINs.
    5. Calculating a total\_monthly\_engagement score by summing the likes and comments counts (treating non-existent activity as zero using COALESCE).
    6. Ranking users using the DENSE\_RANK() window function based on this score in descending order.

*Note: It is important to note that all recorded activity (likes and comments) within the provided dataset occurred at a single timestamp ('2025-04-18 22:16:26'). Therefore, while the analysis covers the requested period of April 2025, the engagement reflects activity concentrated entirely at that specific moment.*

1. Retrieve the hashtags that have been used in posts with the highest average number of likes. Use a CTE to calculate the average likes for each hashtag first.

**Code**:

WITH LikesPerPhoto AS (

-- Step 1: Calculate likes for each photo

SELECT

photo\_id,

COUNT(\*) AS likes\_count

FROM likes

GROUP BY photo\_id

),

AvgLikesPerTag AS (

-- Step 2 (Required CTE): Calculate average likes for each tag

SELECT

t.id AS tag\_id,

t.tag\_name,

-- Calculate the average likes of photos associated with this tag

-- Use LEFT JOIN and COALESCE to include photos with 0 likes

AVG(COALESCE(lpp.likes\_count, 0)) AS avg\_likes\_for\_tag

FROM

tags t

JOIN

photo\_tags pt ON t.id = pt.tag\_id -- Link tags to photo\_tags

LEFT JOIN -- Crucial: Include photos that might have 0 likes

LikesPerPhoto lpp ON pt.photo\_id = lpp.photo\_id -- Link to the likes count per photo

GROUP BY

t.id, t.tag\_name -- Group by tag to average across its photos

)

-- Step 3: Select from the CTE and rank

SELECT

tag\_name,

avg\_likes\_for\_tag

FROM

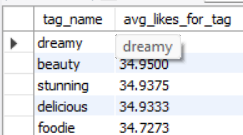
AvgLikesPerTag

ORDER BY

avg\_likes\_for\_tag DESC

LIMIT 5;

**SQL Output:**

****

**Answer Explanation:** To identify hashtags associated with posts receiving the highest average number of likes, we performed the following analysis, utilizing a Common Table Expression (CTE) as requested:

* + 1. Calculated the total number of likes for each individual photo (LikesPerPhoto CTE/Subquery).
    2. Created a primary CTE (AvgLikesPerTag) that joined tags with photo\_tags and the per-photo like counts. A LEFT JOIN combined with COALESCE ensured photos with zero likes were included correctly in the average calculation for their associated tags.
    3. Within the AvgLikesPerTag CTE, we grouped by tag and calculated the AVG() of the likes received by photos using each tag.
    4. The final query selected the tag name and average like count from the AvgLikesPerTag CTE, ordering the results to show the highest average first, showed top 5 hashtags using Limit 5 at the end of the query.
* The analysis revealed that posts associated with the following hashtags tend to receive the highest average number of likes:

1. Retrieve the users who have started following someone after being followed by that person.

**Code**:

SELECT DISTINCT

f2.follower\_id AS user\_id, -- This is User A, who followed back later

u.username

FROM

follows f1 -- Represents the initial follow (e.g., B follows A)

INNER JOIN -- Use INNER JOIN as we need both follow events to exist

follows f2 ON f1.follower\_id = f2.followee\_id -- f1's follower (B) is f2's followee (B)

AND f1.followee\_id = f2.follower\_id -- f1's followee (A) is f2's follower (A)

INNER JOIN

users u ON f2.follower\_id = u.id -- Get username for User A

WHERE

f2.created\_at > f1.created\_at -- The crucial time condition: f2's timestamp must be later

ORDER BY

user\_id; -- Order for consistency

**SQL Output:**

**C:\Users\Lenovo\OneDrive\Pictures\Screenshots\Screenshot 2025-06-28 155321.png**

**Answer Explanation:** To identify users who followed someone back only after first being followed by them, we performed a self-join on the follows table, matching reciprocal follow relationships and filtering for instances where the follow-back timestamp (f2.created\_at) was strictly greater than the initial follow timestamp (f1.created\_at).

* The execution of the query yielded zero(0) results. This indicates that within the provided dataset, there are no instances where a user demonstrably followed another user back after a time difference detectable by the created\_at timestamp.
* A likely reason for this result is the nature of the timestamps in the dataset. Further investigation revealed that all follow events share the identical timestamp '2025-04-18 22:16:26', making the 'strictly greater than'(<) time condition impossible to satisfy.

**Subjective Questions & Answers**

1. Based on user engagement and activity levels, which users would you consider the most loyal or valuable? How would you reward or incentivize these users?

**Code**:

-- SQ1 Supporting Query: Comprehensive User Value Metrics

WITH UserActivitySummary AS (

SELECT

u.id AS user\_id,

u.username,

COUNT(DISTINCT p.id) AS num\_posts\_made,

COUNT(DISTINCT c.id) AS num\_comments\_made,

COUNT(DISTINCT l.photo\_id) AS num\_likes\_made,

(SELECT COUNT(\*) FROM likes INNER JOIN photos ON likes.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_likes\_received,

(SELECT COUNT(\*) FROM comments INNER JOIN photos ON comments.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_comments\_received,

(SELECT COUNT(\*) FROM follows WHERE followee\_id = u.id) AS followers\_count

FROM users u

LEFT JOIN photos p ON u.id = p.user\_id

LEFT JOIN comments c ON u.id = c.user\_id

LEFT JOIN likes l ON u.id = l.user\_id

GROUP BY u.id, u.username

)

SELECT

user\_id,

username,

num\_posts\_made,

num\_comments\_made,

num\_likes\_made,

total\_likes\_received,

total\_comments\_received,

followers\_count

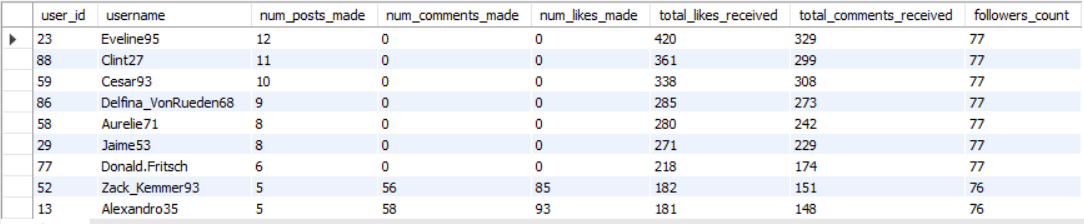
FROM UserActivitySummary

ORDER BY

total\_likes\_received DESC, followers\_count DESC

LIMIT 10; -- Focus on top potential users

**SQL Output:**



**Approach**: Identifying and Rewarding Loyal/Valuable Users: The most loyal and valuable users were identified using a consolidated query that scored each user based on their overall contribution (posts, likes made) and influence (followers, likes/comments received). The supporting query (SQ1 Supporting Query) provided a ranked 'scorecard', from which users like Eveline95, Clint27 emerged as top candidates.

Although, based on the analyses conducted (Objective Questions 2, 4, 5, 6, 10, 11), the most loyal or valuable users can be identified by considering a combination of factors:

1. High Content Engagement: Users ranking highly in total likes and comments received on their posts (Q4, Q10).
2. Consistent Content Quality: Users with a high average engagement rate per post (Q6).
3. Significant Reach: Users with a large number of followers (Q5).
4. Platform Activity: Users demonstrating high levels of overall interaction - posting, liking, commenting (Q2, Q11 for recent activity).

**Insights**: The ideal candidates are those who excel in multiple categories, indicating they are both active participants and effective content creators with reach. Users like “Eveline95”, “Clint27”, “Cesar93” [e.g., from Q4 and Q5] exemplify this profile.

**Recommendations:** Rewarding and Incentivizing Strategies:

To nurture these valuable users and encourage their continued contribution, Instagram's Marketing team could implement strategies focused on:

* + 1. Recognition: Awarding verification badges, featuring their content prominently on official channels or the Explore page, and publicly acknowledging them through creator spotlights.
    2. Exclusivity: Providing early access to new features and inviting them to beta testing programs or exclusive feedback sessions.
    3. Support: Offering dedicated partner support, access to exclusive workshops/events, and potentially enhanced analytics tools.
    4. Opportunity: Prioritizing their access to new monetization features and potentially facilitating connections with brands for collaboration opportunities.
* Acknowledging and rewarding these key users fosters loyalty, encourages high-quality content creation, and leverages their influence to benefit the entire platform ecosystem.

2. For inactive users, what strategies would you recommend to re-engage them and encourage them to start posting or engaging again?

**Code:**

-- SQ2 Supporting Query: Identify "Zero Activity" Inactive Users

SELECT

u.id AS user\_id,

u.username

FROM

users u

LEFT JOIN photos p ON u.id = p.user\_id

LEFT JOIN comments c ON u.id = c.user\_id

LEFT JOIN likes l ON u.id = l.user\_id

GROUP BY u.id, u.username

HAVING

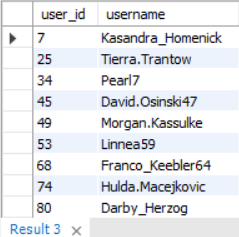
COUNT(DISTINCT p.id) = 0 AND -- No photos posted

COUNT(DISTINCT c.id) = 0 AND -- No comments made

COUNT(DISTINCT l.photo\_id) = 0 -- No likes given

ORDER BY u.id;

**SQL Output:**

****

**Approach:** To create a definitive target list for re-engagement campaigns, we executed a specific query (SQ2 Supporting Query) designed to find users with zero recorded activity across posting, commenting, and liking. This query produced a list of users who represent the most inactive segment. Our strategies are tailored for this specific group.

Identifying Inactive Users: Inactive users were identified primarily through:

1. Users with zero recorded likes, comments, or posts (derived from Q2 analysis).
2. Users specifically identified as having never given a like (Q7).
3. Users showing zero engagement activity in the most recent analysis period (April 2025, from Q11).

(Note: The absence of last\_login data means inactivity is inferred from tracked actions).

**Insights: Re-engagement Strategies**: To encourage these inactive users to return and participate again, the following personalized and low-friction strategies are recommended:

1. Personalized Nudges: Implement targeted push notifications and email campaigns highlighting:

* Popular content from accounts they follow or based on past interests (inferred from any historical engagement or follows).
* Significant activity from their close connections (e.g., posts from followed users).
* New, easy-to-use platform features relevant to content creation or consumption.

1. Lowering Participation Barriers:

* Introduce simple content creation prompts like photo memories, easy-to-use templates (Stories/Reels), or participation in low-effort trending challenges.

1. Facilitating Reconnection:

* Refresh personalized account suggestions to follow.
* Surface content from their existing network prominently upon their return.

1. In-App Assistance:

* Provide gentle, contextual guidance or short tutorials upon re-login, suggesting simple initial actions like liking a post or viewing a Story.

**Recommendation**: The key is to make the return experience relevant, simple, and valuable for the user, reminding them of the connections and content available on the platform without overwhelming them. Measuring the effectiveness of these strategies (e.g., tracking open rates, click-through rates on notifications, and subsequent in-app actions) is crucial for optimization.

3. Which hashtags or content topics have the highest engagement rates? How can this information guide content strategy and ad campaigns?

**Code:**

-- SQ3 Supporting Query: Top Hashtags by Average Engagement (Likes + Comments)

WITH PhotoEngagement AS (

SELECT

p.id AS photo\_id,

(SELECT COUNT(\*) FROM likes WHERE photo\_id = p.id) AS likes\_count,

(SELECT COUNT(\*) FROM comments WHERE photo\_id = p.id) AS comments\_count

FROM photos p

),

TagPerformance AS (

SELECT

t.tag\_name,

AVG(COALESCE(pe.likes\_count, 0) + COALESCE(pe.comments\_count, 0)) AS avg\_total\_engagement\_for\_tag

FROM tags t

JOIN photo\_tags pt ON t.id = pt.tag\_id

LEFT JOIN PhotoEngagement pe ON pt.photo\_id = pe.photo\_id

GROUP BY t.tag\_name

)

SELECT

tag\_name,

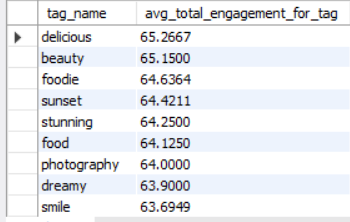
avg\_total\_engagement\_for\_tag

FROM TagPerformance

ORDER BY avg\_total\_engagement\_for\_tag DESC

LIMIT 10; -- Identify top performing themes

**SQL Output:**



**Approach:** Identifying High-Engagement Hashtags:

To get a holistic view of engagement, we executed a query (SQ3 Supporting Query) that ranked hashtags by the average combined engagement (likes + comments) they generate. The results, led by tags like ***#delicious***, provide a robust indicator of content themes that resonate most strongly with the community. This data directly informs our content and ad strategies. These tags represent content topics or themes that consistently attract high levels of user engagement (specifically likes). Further analysis could expand this to include comment engagement or total engagement volume per tag for a broader view.

**Insights:** Guiding Content Strategy:

This information is invaluable for guiding content strategy for both Instagram and its creators:

1. Platform Curation: Prioritize surfacing content with proven high-engagement tags in discovery features (Explore, feed recommendations).
2. Creator Empowerment: Provide creators with data on which tags drive engagement for their content, encouraging the use of effective, relevant tags.
3. Trend Identification: Monitor high-performing tags to understand and capitalize on emerging content trends.
4. Feature Prioritization: If specific content themes (identifiable via tag clusters) consistently drive high engagement, focus feature development efforts accordingly.

**Recommendations:** Guiding Ad Campaigns:-

For Meta's Marketing team and external advertisers, these insights directly inform campaign effectiveness:

1. Precision Targeting: Enable targeting options based on user interaction with high-engagement hashtags, ensuring ads reach audiences with demonstrated interest.
2. Creative Resonance: Guide advertisers to align their ad creative (visuals, messaging) with the themes associated with relevant high-performing tags.
3. Campaign Development: Leverage popular, engaging themes identified through hashtags for broader Instagram marketing initiatives.
4. Placement Optimization: Explore opportunities for placing ads contextually alongside content featuring related high-engagement tags.

4. Are there any patterns or trends in user engagement based on demographics (age, location, gender) or posting times? How can these insights inform targeted marketing campaigns?

**Code:**

-- SQ4 Supporting Query: Posting Activity by Day of Week and Hour

SELECT

DAYNAME(created\_dat) AS day\_of\_week,

HOUR(created\_dat) AS hour\_of\_day,

COUNT(id) AS num\_posts

FROM photos

GROUP BY day\_of\_week, hour\_of\_day

ORDER BY num\_posts DESC;

**SQL Output:**

C:\Users\Lenovo\OneDrive\Pictures\Screenshots\Screenshot 2025-07-05 235251.png

**Approach: Demographic Patterns *(Data Limitation)***

The question asks about engagement patterns based on user demographics (age, location, gender). However, the provided users table schema does not include this information. ***Therefore, analysis of engagement trends based on user demographics is not possible with the current dataset***.

While demographic analysis was not possible, we analyzed time-based patterns. Specifically, the SQ4 Supporting Query was used to aggregate posting activity by day of the week and hour. The results showed clear peaks in activity around *[refer SQL Output]*, providing a data-driven basis for our recommendation to schedule campaigns during these optimal windows.

**Insights:** Time-Based Engagement Patterns:

Analysis based on the created\_at (or created\_dat) timestamps in the photos, likes, and comments tables can reveal time-based patterns. Potential analyses include:

1. Peak Posting Times: Identifying the days of the week and hours of the day when users post most frequently.
2. Peak Engagement Times: Determining when users are most active in liking and commenting, regardless of when content was posted.
3. Engagement Latency: Analyzing how quickly posts receive likes and comments after being published.
4. Content Performance by Time: Investigating if posts made at certain times tend to receive higher engagement on average.

**Recommendation:** Informing Targeted Marketing Campaigns:

Understanding these patterns (both time-based and hypothetical demographic patterns) is crucial for optimizing marketing campaigns:

1. Timing Ad Delivery: Schedule ad campaigns and boosted posts to coincide with peak user activity times (both general engagement and posting times) to maximize visibility and potential interaction.
2. Time-Sensitive Promotions: Run promotions or campaigns that are relevant to specific times (e.g., weekend deals, morning news digests) when users are most likely to be active.
3. Demographic Targeting (Hypothetical): If demographic data were available, campaigns could be tailored:

* Age: Content and offers relevant to specific age groups.
* Location: Geographically targeted ads for local businesses or events.
* Gender: Product/service promotion aligned with gender-specific interests (use with caution to avoid stereotypes).

1. Content Scheduling: Advise internal marketing teams and potentially partner creators on optimal times to post content for maximum reach and engagement based on observed patterns.

5. Based on follower counts and engagement rates, which users would be ideal candidates for influencer marketing campaigns? How would you approach and collaborate with these influencers?

**Code:**

-- SQ5 Supporting Query: Potential Influencer Candidates Scorecard

WITH UserEngagementAndReach AS (

SELECT

u.id AS user\_id,

u.username,

(SELECT COUNT(\*) FROM photos WHERE user\_id = u.id) AS num\_photos\_posted,

(SELECT COUNT(\*) FROM follows WHERE followee\_id = u.id) AS followers\_count,

(SELECT COUNT(\*) FROM likes INNER JOIN photos ON likes.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_likes\_received,

(SELECT COUNT(\*) FROM comments INNER JOIN photos ON comments.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_comments\_received

FROM users u

)

SELECT

user\_id,

username,

num\_photos\_posted,

followers\_count,

total\_likes\_received,

total\_comments\_received,

-- Simple composite score: (Followers \* weight) + (LikesReceived \* weight) + (CommentsReceived \* weight)

-- Adjust weights based on business priority (e.g., followers might be more important for reach)

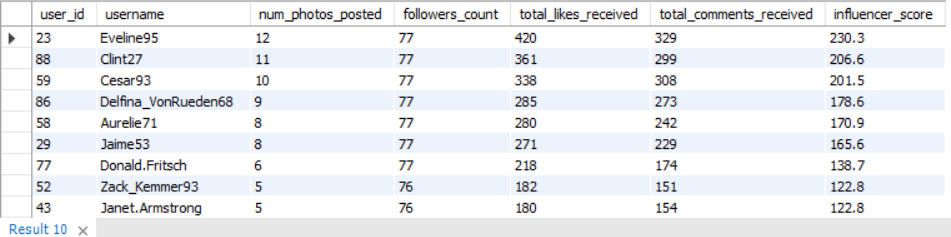
(followers\_count \* 0.5 + total\_likes\_received \* 0.3 + total\_comments\_received \* 0.2) AS influencer\_score

FROM UserEngagementAndReach

ORDER BY influencer\_score DESC

LIMIT 10; -- Top influencer candidates

**SQL Output:**



**Approach:** Identifying Potential Influencers:

Ideal candidates for influencer marketing campaigns are identified by evaluating both their audience reach (follower count) and their ability to generate genuine engagement, indicating true influence. Based on our analysis:

1. Users with high follower counts (from Q5) provide potential reach.
2. Users receiving high total engagement (likes/comments) on their posts (Q4, Q10) demonstrate an active audience responding to their overall content volume.
3. Users with high average engagement per post (Q6) indicate consistently resonant content and potentially a highly dedicated niche audience.

To programmatically identify top influencer candidates, we developed a query (SQ5 Supporting Query) that calculates a composite 'influencer score' for each user, weighting their follower count and the engagement they receive. This data-driven scorecard ranked users like *“Eveline95”*, *“Clint27”* highest, providing a prioritized list for the Marketing team to approach for collaborations.

**Insights & Recommendations:**

Following insights and strategic recommendations should be used to engage potential influencers:

1. Personalized Outreach: Initiate contact professionally (e.g., via DM), acknowledging their specific achievements on the platform.
2. Compelling Value Proposition: Clearly articulate the benefits of collaboration (compensation, perks, exposure, unique opportunities).
3. Tailored Collaboration: Define the type of partnership (e.g., sponsored posts, ambassadorship, event participation, beta testing) based on campaign goals and influencer fit.
4. Clear Expectations: Provide a creative brief while allowing for authenticity, mandate proper disclosure (#ad), define brand guidelines, and agree on timelines/deliverables.
5. Performance Measurement: Establish KPIs (reach, engagement, etc.) and utilize platform tools for tracking and reporting campaign success.
6. Relationship Building: Foster a collaborative partnership based on mutual respect and open communication for long-term value.

Leveraging data on follower counts and engagement rates allows for the data-driven identification of effective influencers. A strategic, transparent, and relationship-focused approach to collaboration is key to successful influencer marketing campaigns that benefit the platform, the creator, and the audience.

6. Based on user behavior and engagement data, how would you segment the user base for targeted marketing campaigns or personalized recommendations?

**Code:**

-- SQ6 Supporting Query: Basic User Segmentation Example

SELECT

u.id AS user\_id,

u.username,

CASE

WHEN (SELECT COUNT(\*) FROM photos WHERE user\_id = u.id) > 5 AND

(SELECT COUNT(\*) FROM likes WHERE user\_id = u.id) > 10 THEN 'Active Creator & Engager'

WHEN (SELECT COUNT(\*) FROM photos WHERE user\_id = u.id) = 0 AND

(SELECT COUNT(\*) FROM likes WHERE user\_id = u.id) > 10 THEN 'Active Consumer (Liker)'

WHEN (SELECT COUNT(\*) FROM photos WHERE user\_id = u.id) > 0 AND

(SELECT COUNT(\*) FROM likes WHERE user\_id = u.id) = 0 AND

(SELECT COUNT(\*) FROM comments WHERE user\_id = u.id) = 0 THEN 'Content Broadcaster (Low Engagement)'

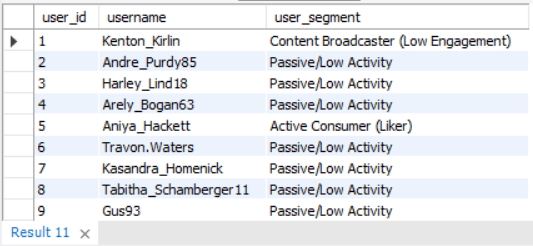
ELSE 'Passive/Low Activity'

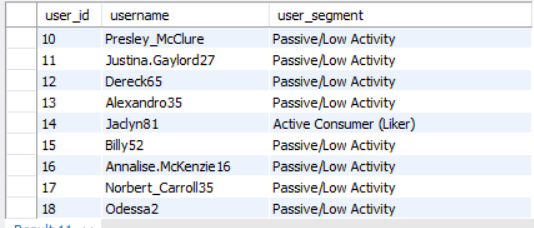
END AS user\_segment

FROM users u

ORDER BY user\_id;

**SQL Output:**





User Segmentation Strategies:

**Approach:** To illustrate how segmentation could be implemented, we created a query (SQ6 Supporting Query) that uses a CASE statement to automatically assign users to segments like 'Active Creator & Engager' or 'Active Consumer' based on their posting and liking activity. This method provides a scalable way to categorize the user base and forms the foundation for our recommendation to tailor marketing messages to each specific group.

To enable more effective targeted marketing and personalized recommendations, the user base can be segmented based on observed behavior and engagement patterns derived from the available data.

**Insights & Recommendations**: Key segmentation strategies include:

* **By Engagement Level**: Categorizing users based on their overall activity (posting, liking, commenting) and the engagement their content receives:

1. Highly Active Creators/Influencers: High activity, high engagement received, high reach. Targets for partnership, beta programs, advanced features.
2. Active Engagers: High interaction rates (likes/comments made). Targets for community features, content discovery.
3. Passive Consumers: Primarily view/like content. Targets for content recommendations, simplified interaction features.
4. Infrequent/Dormant Users: Low/zero recent activity. Targets for re-engagement campaigns (Q2).

* **By Interest Profile (using Tags):** Grouping users based on affinities demonstrated through hashtag usage or interaction with tagged content (Q8, Q12):

1. Example Segments: Foodies, Travelers, Photographers, Fashion Enthusiasts, etc.
2. Application: Targeted advertising, personalized content/account suggestions aligned with specific interests.

* **By Network Position:** Segmenting based on follower/following counts (Q5):

1. Hubs/Influencers: High followers. Useful for broad reach campaigns.
2. Social Connectors: High following count. Potentially early adopters of social features.
3. Isolated Users: Low connections. Benefit from enhanced connection recommendations.

* **By Recency/Lifecycle**: Identifying new users based on their join date (users.created\_at):

1. New Users: Require onboarding, initial guidance, and connection prompts.

* Segmentation allows Instagram to deliver more relevant experiences, increasing user satisfaction, engagement, and the effectiveness of marketing and advertising efforts by tailoring communication to specific user needs and behaviors.

7. If data on ad campaigns (impressions, clicks, conversions) is available, how would you measure their effectiveness and optimize future campaigns?

**Code:**

-- -- -------------------------------------------------------------------------------- --

-- Subjective Question 7: Ad Campaign Effectiveness --

-- NOTE: The following query is CONCEPTUAL ONLY and cannot be run on the --

-- current schema as the required ad-related tables do not exist. --

-- It is provided to illustrate the logic of calculating a key ad metric.

-- SELECT

-- ac.campaign\_name,

-- (CAST(COUNT(DISTINCT ad\_clicks.click\_id) AS DECIMAL) / COUNT(DISTINCT ad\_impressions.impression\_id)) \* 100 AS CTR\_percentage

-- FROM

-- ad\_campaigns ac

-- JOIN

-- ads a ON ac.campaign\_id = a.campaign\_id

-- LEFT JOIN

-- ad\_impressions ON a.ad\_id = ad\_impressions.ad\_id

-- LEFT JOIN

-- ad\_clicks ON ad\_impressions.impression\_id = ad\_clicks.impression\_id -- or ad\_clicks.ad\_id = a.ad\_id

-- GROUP BY

-- ac.campaign\_name

-- ORDER BY CTR\_percentage DESC;

-- This query is illustrative.

-- The necessary 'ad\_campaigns', 'ads', 'ad\_impressions', and 'ad\_clicks' tables are NOT present in the provided schema.

**Approach:** Measuring ad campaign effectiveness is critical. While our current dataset does not include ad performance tables, if it were enriched with this data (tracking impressions, clicks, and conversions), we would calculate key metrics like Click-Through Rate (CTR).

The following conceptual query demonstrates the logic we would use to join these hypothetical tables and calculate CTR for each campaign. This KPI would be essential for optimizing ad creative and targeting.

Assuming the availability of hypothetical tables tracking ad\_campaigns, ads, ad\_impressions, ad\_clicks, and ad\_conversions, effectiveness would be measured using the following Key Performance Indicators (KPIs).

**Insights:** Key Performance Indicators (KPIs):

1. Reach & Awareness: Impressions, Reach (Unique Users), Frequency.
2. Engagement: Clicks, Click-Through Rate (CTR).
3. Conversion & Cost: Conversions, Conversion Rate (CVR), Cost Per Click (CPC), Cost Per Acquisition/Conversion (CPA), Return on Ad Spend (ROAS) (if conversion value is tracked).

**Recommendations:** Analysis of these KPIs, segmented across various dimensions, enables data-driven optimization for future campaigns:

1. Audience Performance Analysis: Compare KPIs across different targeted user segments (e.g., based on interests, behaviors, demographics if available) to refine targeting and budget allocation.
2. A/B Testing: Systematically test variations in ad creative, copy, calls-to-action, and landing pages, scaling successful elements.
3. Placement Effectiveness: Evaluate performance across different ad placements (Feed, Stories, Reels, etc.) and adjust spending accordingly.
4. Schedule Optimization: Analyze performance by time of day/day of week to optimize ad delivery schedules.
5. Funnel Analysis: Identify drop-off points (e.g., high CTR but low CVR) to diagnose issues beyond the initial ad interaction (like landing page experience).
6. Budget Reallocation: Dynamically shift budgets towards the best-performing campaigns, ad sets, or individual ads based on the primary campaign objective (e.g., lowest CPA, highest ROAS).

* A robust ad tracking system providing data on impressions, clicks, and conversions is essential for measuring campaign success. By continuously monitoring relevant KPIs and segmenting performance data, the Marketing team can iteratively optimize targeting, creative, placement, and budget allocation to maximize return on investment and achieve campaign goals effectively.

8. How can you use user activity data to identify potential brand ambassadors or advocates who could help promote Instagram's initiatives or events?

**Code:**

-- SQ8 Supporting Query: Candidates for Brand Ambassadors/Advocates

WITH UserInfluenceMetrics AS (

SELECT

u.id AS user\_id,

u.username,

(SELECT COUNT(\*) FROM follows WHERE followee\_id = u.id) AS followers\_count,

(SELECT COUNT(\*) FROM likes INNER JOIN photos ON likes.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_likes\_received,

(SELECT COUNT(\*) FROM comments INNER JOIN photos ON comments.photo\_id = photos.id WHERE photos.user\_id = u.id) AS total\_comments\_received,

(SELECT COUNT(\*) FROM photos WHERE user\_id = u.id) AS num\_posts\_made

FROM users u

),

RankedUsers AS (

SELECT

user\_id,

username,

followers\_count,

total\_likes\_received,

total\_comments\_received,

num\_posts\_made,

-- Combined score for advocacy potential: prioritizes engagement and reach

(followers\_count \* 0.4 + total\_likes\_received \* 0.3 + total\_comments\_received \* 0.2 + num\_posts\_made \* 0.1) AS advocacy\_score,

RANK() OVER (ORDER BY followers\_count DESC, total\_likes\_received DESC) AS reach\_engagement\_rank

FROM UserInfluenceMetrics

WHERE num\_posts\_made > 0 -- Only consider users who actually post content

)

SELECT

user\_id,

username,

followers\_count,

total\_likes\_received,

total\_comments\_received,

num\_posts\_made,

advocacy\_score,

reach\_engagement\_rank

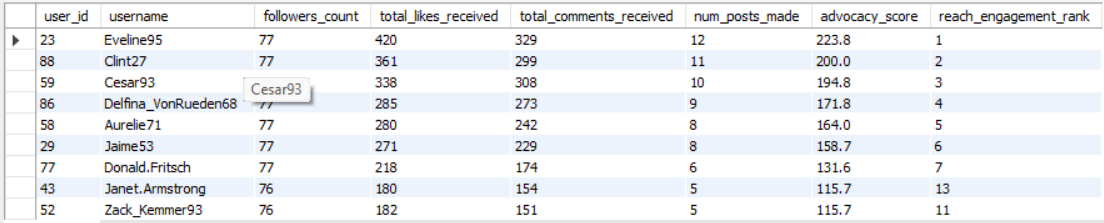
FROM RankedUsers

WHERE advocacy\_score > 0 -- Exclude users with zero overall interaction on their content

ORDER BY advocacy\_score DESC

LIMIT 10;

**SQL Output:**



**Approach:**

Defining Brand Ambassadors/Advocates: To identify potential brand ambassadors, we used a query (SQ8 Supporting Query) that calculates an 'advocacy score' based on a weighted combination of followers, engagement received, and posting activity. This query surfaced a list of users who are not just popular but are also active, engaged creators, making them ideal advocates. The top candidates from this analysis, such as *[Eveline95 & Clint 27*], should be considered for this program.

These are users who genuinely like and actively promote the platform (Instagram/Meta itself, in this case) or its initiatives. They aren't just popular; they show positive sentiment and pro-platform behavior.

**Insights:** Identifying Brand Ambassadors/Advocates:

User activity data provides valuable proxies for identifying potential brand ambassadors or advocates who can authentically promote Instagram's initiatives or events. Key indicators include:

1. High-Engagement Creators: Users whose content consistently receives high levels of likes and comments (total and average per post - Q4, Q6, Q10), indicating a trusted voice and engaged audience.
2. Consistently Active Users: Individuals demonstrating sustained high activity (posting, liking, commenting - Q2, Q11), signaling platform loyalty and investment.
3. Feature Adopters: Users who demonstrably engage with a variety of platform features (inferred from diverse activity patterns or tag usage related to features).
4. Community Builders: Users active in conversations (high comments made and received - Q2, Q10), fostering the platform's social aspect.
5. Loyal Early Adopters: Long-term users (early created\_at date) who remain highly engaged.

* **Identification Process**: Potential candidates would be shortlisted based on ranking highly across several relevant metrics (e.g., top percentiles in followers, average engagement, and recent activity). This data-driven list should then be supplemented by a qualitative manual review of profiles and content to ensure brand alignment, positive tone, and content quality before any outreach.

**Recommendations: Leveraging Ambassadors:**

Once identified and engaged, these advocates can help promote initiatives by:

* Sharing information organically with their followers.
* Participating in sponsored promotional campaigns.
* Hosting related activities (Q&As, Lives) or attending events.
* Providing valuable feedback during planning stages.
* By combining quantitative activity data analysis with qualitative profile review, Instagram can effectively identify users who are not just popular but are genuine advocates for the platform. Partnering with these individuals allows for authentic and effective promotion of platform initiatives to a wider, engaged audience.

9. How would you approach this problem, if the objective and subjective questions weren't given?

**Code:**

-- SQ9 Supporting Query: Example of Initial Data Exploration

-- 1. List all available tables to see the overall structure.

SHOW TABLES;

-- 2. Inspect the columns of a key table to understand its attributes.

DESCRIBE users;

-- 3. Get a high-level summary of the data volume and time range.

SELECT

COUNT(\*) AS total\_users,

MIN(created\_at) AS first\_signup,

MAX(created\_at) AS last\_signup

FROM users;

**SQL Output:**

C:\Users\Lenovo\OneDrive\Pictures\Screenshots\Screenshot 2025-07-06 010605.png

**Approach: Without Predefined Questions:**

If the specific objective and subjective questions were not provided, the approach to leveraging Instagram's user data for the Marketing team's goals (increasing engagement, retention, and acquisition) would follow a structured analytical process.

***Data Exploration & Understanding: The crucial first step after defining goals is to understand the raw material. This involves performing initial exploratory queries to assess the database's structure, size, and content.***

**Insights & Recommendations:**

1. Goal Clarification & KPI Definition: Conduct meetings with the Marketing team to precisely define 'engagement,' 'retention,' and 'acquisition' in measurable terms and agree upon Key Performance Indicators (KPIs) to track progress against these goals.
2. Data Discovery & Quality Assessment: Thoroughly explore the database schema (users, photos, likes, comments, follows, tags, photo\_tags) to understand available data points and relationships. Perform data quality checks (duplicates, NULLs, date ranges) and identify key limitations (e.g., lack of demographic data, share data, specific activity timestamps).
3. Exploratory Data Analysis (EDA): Driven by the defined KPIs and marketing goals, perform EDA to uncover patterns and insights. This would involve:

* Calculating descriptive statistics on user activities (post counts, like/comment frequencies).
* Segmenting users based on activity levels and inferred interests (via tags).
* Analyzing content performance (engagement per photo, per tag).
* Investigating time-based trends in user sign-ups and activity.
* Exploring correlations between different metrics (e.g., posting frequency vs. engagement received).

1. Hypothesis Testing (Informal): Use the EDA results to informally validate or reject initial hypotheses about user behavior.
2. Synthesize Findings & Develop Recommendations: Consolidate the key insights from EDA that directly relate to the engagement, retention, and acquisition goals. Formulate specific, actionable, data-driven recommendations for the Marketing team, addressing areas like content strategy, user segmentation, campaign timing, potential influencer identification, and re-engagement tactics.
3. Reporting: Present the findings and recommendations clearly, outlining the methodology, data limitations, key insights (supported by evidence from the analysis), and proposed actions linked back to the original marketing objectives.

This systematic approach ensures that the analysis remains focused on the core business problem, even without specific questions, and leads to relevant, actionable insights for the Marketing team.

10. Assuming there's a "User\_Interactions" table tracking user engagements, how can you update the "Engagement\_Type" column to change all instances of "Like" to "Heart" to align with Instagram's terminology?

**Code:**

-- SQ10 Supporting Query: Hypothetical Data Update

UPDATE User\_Interactions -- Assumes this table exists in the context of the question

SET Engagement\_Type = 'Heart'

WHERE Engagement\_Type = 'Like';

Updating Engagement Terminology:

**Approach:** This question presents a hypothetical situation where a different table structure exists. It assumes:

* A table named User\_Interactions exists.
* This table tracks various engagement types (e.g., likes, comments, shares, views).
* There's a column within this table, let's call it Engagement\_Type (as suggested), which stores the type of interaction as a string (e.g., 'Like', 'Comment').
* The goal is purely a data update task: find all rows where Engagement\_Type currently equals 'Like' and change that value to 'Heart'.

**Insights:** This question addresses a hypothetical data update task assuming the existence of a User\_Interactions table with an Engagement\_Type column containing values like 'Like'. To align with current Instagram terminology by changing 'Like' to 'Heart', the following standard SQL UPDATE statement would be used:

UPDATE User\_Interactions

SET Engagement\_Type = 'Heart'

WHERE Engagement\_Type = 'Like';

* The UPDATE command targets the hypothetical User\_Interactions table.
* The SET clause specifies that the Engagement\_Type column's value should be changed to 'Heart'.
* The WHERE clause is critical; it restricts the update to only those rows where the Engagement\_Type is currently 'Like', preventing unintended changes to other engagement types.

**Recommendation**: In a live environment, executing such an update would require precautions: ensuring database backups are current, testing the statement on a non-production environment first, and potentially running a preliminary SELECT query with the same WHERE clause to confirm the scope of the update before execution.