**Social Media Analysis**

**Tasks**

**Objective Questions**

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

By running the below queries, wo got to know that there are no duplicates in the tables.

SELECT comment\_text, user\_id, photo\_id, created\_at

FROM comments

GROUP BY 1,2,3,4

HAVING COUNT(\*) > 1;

SELECT follower\_id, followee\_id

FROM follows

GROUP BY 1,2

HAVING COUNT(\*) > 1;

SELECT user\_id, photo\_id

FROM likes

GROUP BY 1,2

HAVING COUNT(\*) > 1;

SELECT photo\_id, tag\_id

FROM photo\_tags

GROUP BY 1,2

HAVING COUNT(\*) > 1;

SELECT image\_url, user\_id, created\_dat

FROM photos

GROUP BY 1,2,3

HAVING COUNT(\*) > 1;

SELECT tag\_name, created\_at

FROM tags

GROUP BY 1,2

HAVING COUNT(\*) > 1;

SELECT username, created\_at

FROM users

GROUP BY 1,2

HAVING COUNT(\*) > 1;

The above queries were used to find the null values. We received no null values in any of the tables.

SELECT \* FROM comments

WHERE id IS NULL OR comment\_text IS NULL OR user\_id IS NULL OR photo\_id IS NULL OR created\_at IS NULL;

SELECT \* FROM follows

WHERE follower\_id IS NULL OR followee\_id IS NULL OR created\_at IS NULL;

SELECT \* FROM likes

WHERE user\_id IS NULL OR photo\_id IS NULL OR created\_at IS NULL;

SELECT \* FROM photo\_tags

WHERE photo\_id IS NULL OR tag\_id IS NULL;

SELECT \* FROM photos

WHERE id IS NULL OR image\_url IS NULL OR user\_id IS NULL OR created\_dat IS NULL;

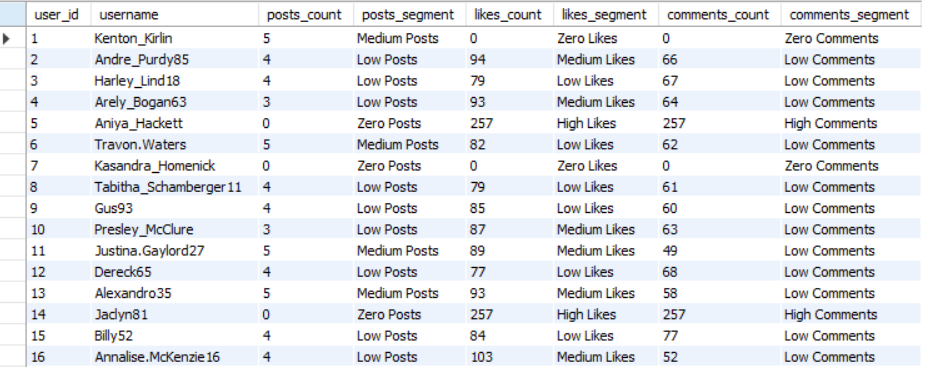
SELECT \* FROM tags

WHERE id IS NULL OR tag\_name IS NULL OR created\_at IS NULL;

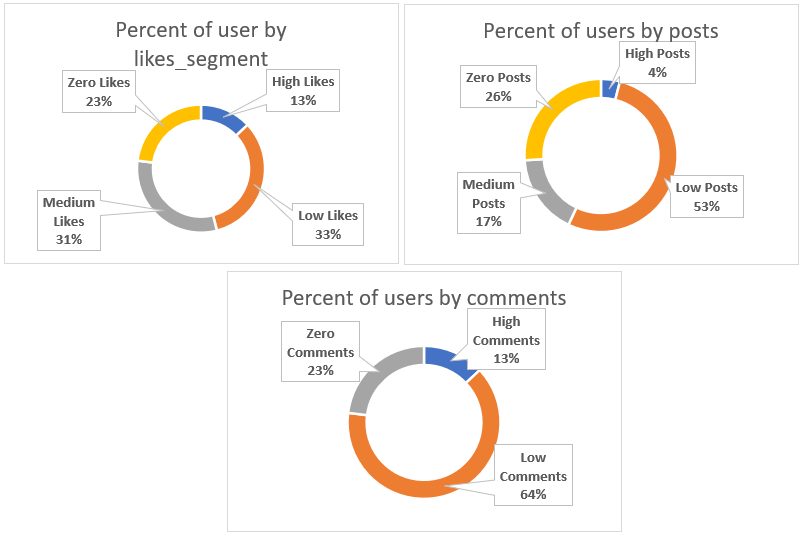
SELECT \* FROM users

WHERE id IS NULL OR username IS NULL OR created\_at IS NULL;

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



The above table shows the user id, username, post count, likes count, comments count and their segment.



The above charts show the distribution of user activity level (posts, likes and comments).

The order of segments can be observed as –

1. **Likes Segment –**

Low Likes (33%) > Medium Likes (31%) > Zero Likes (23%) > High Likes (13%)

1. **Posts Segment –**

Low Posts (53%) > Zero Posts (26%) > Medium Posts (17%) > High Posts (4%)

1. **Comments Segment –**

Low Comments (64%) > Zero Comments (23%) > High Comments (13%)

The below query was used to retrieve the information and then the result was sent to excel. In excel, result was used to create pivot table and chart to create the above doughnut charts.

SELECT u.id AS user\_id, u.username,

COUNT(DISTINCT p.id) AS posts\_count,

CASE

WHEN COUNT(DISTINCT p.id) = 0 THEN 'Zero Posts'

WHEN COUNT(DISTINCT p.id) <= MAX(COUNT(DISTINCT p.id)) OVER() / 3 THEN 'Low Posts'

WHEN COUNT(DISTINCT p.id) <= 2 \* MAX(COUNT(DISTINCT p.id)) OVER() / 3 THEN 'Medium Posts'

ELSE 'High Posts'

END AS posts\_segment,

COUNT(DISTINCT l.photo\_id) AS likes\_count,

CASE

WHEN COUNT(DISTINCT l.photo\_id) = 0 THEN 'Zero Likes'

WHEN COUNT(DISTINCT l.photo\_id) <= MAX(COUNT(DISTINCT l.photo\_id)) OVER() / 3 THEN 'Low Likes'

WHEN COUNT(DISTINCT l.photo\_id) <= 2 \* MAX(COUNT(DISTINCT l.photo\_id)) OVER() / 3 THEN 'Medium Likes'

ELSE 'High Likes'

END AS likes\_segment,

COUNT(DISTINCT c.id) AS comments\_count,

CASE

WHEN COUNT(DISTINCT c.id) = 0 THEN 'Zero Comments'

WHEN COUNT(DISTINCT c.id) <= MAX(COUNT(DISTINCT c.id)) OVER() / 3 THEN 'Low Comments'

WHEN COUNT(DISTINCT c.id) <= 2 \* MAX(COUNT(DISTINCT c.id)) OVER() / 3 THEN 'Medium Comments'

ELSE 'High Comments'

END AS comments\_segment

FROM users u

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

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

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

GROUP BY 1, 2;

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

****

The average number of tags per post = 1.95 (rounded to 2 decimal places).

The query used to get the output –

SELECT ROUND(AVG(tags\_per\_post), 2) AS avg\_num\_tags\_per\_post

FROM (

SELECT p.id AS post, COUNT(pt.tag\_id) AS tags\_per\_post

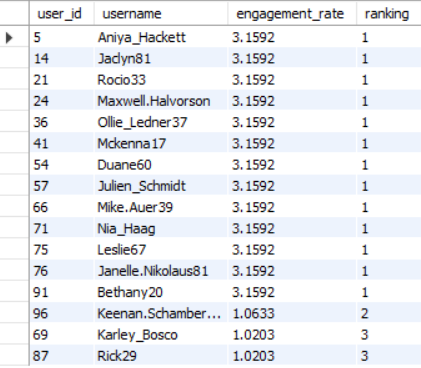
FROM photos p

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

GROUP BY 1

) dt;

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

****

The table above shows the engagement rate (likes + comments) and we have ranked users based on the engagement rate in the ‘ranking’ column. We have taken the highest engagement rate (ranking up to 3).

13 users have highest engagement rate of 3.1592.

The query used to get the output is –

WITH cte AS (

SELECT u.id AS user\_id, u.username,

(COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) AS engagement,

(COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) \* 100 / SUM((COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id))) OVER() AS engagement\_rate,

DENSE\_RANK() OVER(ORDER BY (COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) DESC) AS highest\_engagements

FROM users u

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

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

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

GROUP BY 1

)

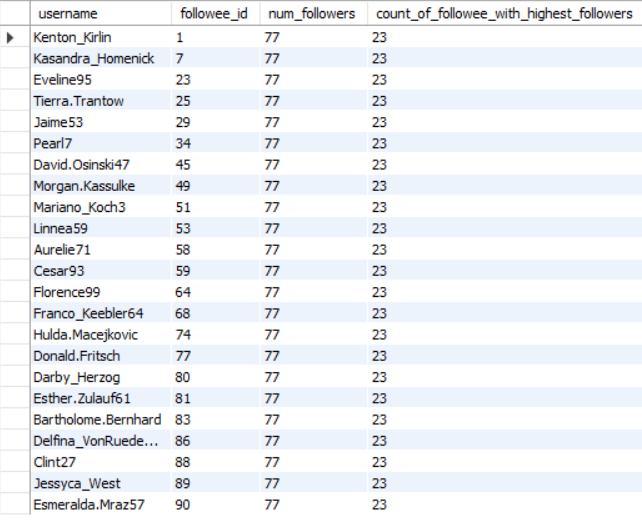
SELECT user\_id, username, engagement\_rate, highest\_engagements AS ranking

FROM cte

WHERE highest\_engagements IN (1, 2, 3)

ORDER BY 4, 1;

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

****

The above table shows the name of users with highest number of followers.

The query used to get the output is –

WITH cte AS (

SELECT u.username, f.followee\_id, COUNT(follower\_id) AS num\_followers

FROM follows f

JOIN users u ON u.id = f.followee\_id

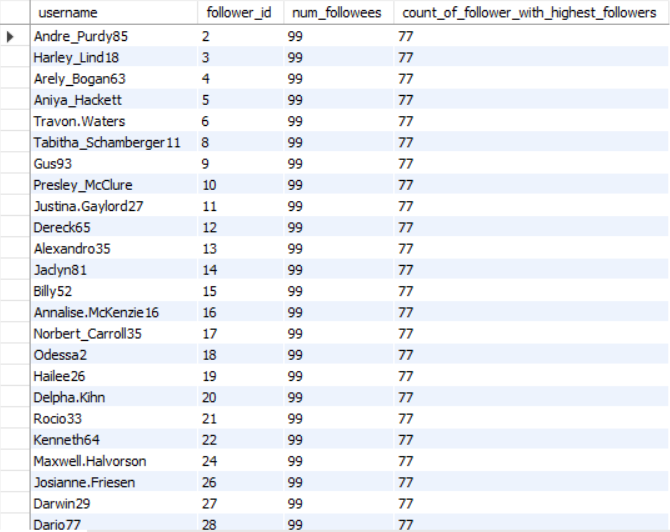
GROUP BY 1, 2

)

SELECT username, followee\_id, num\_followers, COUNT(followee\_id) OVER() AS count\_of\_followee\_with\_highest\_followers

FROM cte

WHERE num\_followers = (SELECT MAX(num\_followers) FROM cte);

****

The above table snippet shows the name of the users with highest number of followees.

The query used to get the output is –

WITH cte AS (

SELECT u.username, follower\_id, COUNT(followee\_id) AS num\_followees

FROM follows f

JOIN users u ON u.id = f.follower\_id

GROUP BY 1, 2

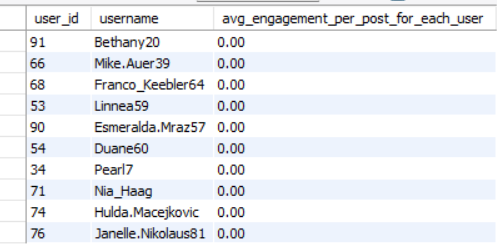
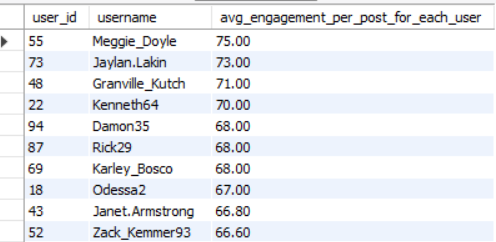
)

SELECT username, follower\_id, num\_followers, COUNT(follower\_id) OVER() AS count\_of\_follower\_with\_highest\_followers

FROM cte

WHERE num\_followees = (SELECT MAX(num\_followees) FROM cte);

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

****

The above tables shows the average engagement rate per post per users for for top 10 users and bottom 10 users.

The query used to get the output –

WITH cte AS (

SELECT u.id AS user\_id, u.username, p.id AS post\_id, (COUNT(DISTINCT l.user\_id) + COUNT(DISTINCT c.id)) AS total\_engagement

FROM users u

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

LEFT JOIN likes l ON P.ID = l.photo\_id

LEFT JOIN comments c ON P.ID = c.photo\_id

GROUP BY 1, 2, 3

)

SELECT DISTINCT user\_id, username,

ROUND(AVG(total\_engagement) OVER(PARTITION BY user\_id), 2) AS avg\_engagement\_per\_post\_for\_each\_user

FROM cte

ORDER BY 3 DESC;

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

****

****

The above table shows the list of users who have never liked a post.

The query used to get the output –

SELECT username

FROM (

SELECT u.id as user\_id, u.username, COUNT(DISTINCT l.photo\_id) AS likes\_count

FROM users u

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

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

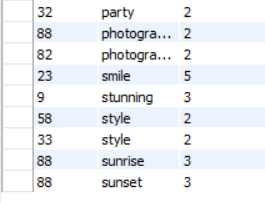
GROUP BY 1, 2

HAVING COUNT(DISTINCT l.photo\_id) = 0

) dt

ORDER BY 1;

1. **How can you leverage user-generated content (posts, hashtags, photo tags) to create more personalized and engaging ad campaigns?**

****

The above table shows the user id, tag name and how many times that tag is used by a user. This will help us know what kind of tags user generally use and would give us an idea of his preferences.

The chart above shows the sum of count of tags used by users. From this chart we can observe that –

1. Party is the mostly used tags by the users.
2. Other popular tags include food, fashion and beauty.
3. The least used tags are foodie and landscape.
4. Food and foodie are similar tags but food is used more than foodie. This can help us **use** **tags that are popular among the users**.

Steps for using this data for creating more personalized and engaging campaigns –

1. **Targeted Ads:** Show ads that align with users’ favorite topics (e.g., food, fashion).
2. **Interest Groups:** Group users by similar tags to deliver more relevant ads.
3. **Optimized Ads:** Use popular tags in ad messaging to make ads more engaging.

The query used to get the output –

SELECT id AS user\_id, tag\_name, tags\_count

FROM (

SELECT u.id, t.tag\_name, COUNT(t.tag\_name) AS tags\_count,

DENSE\_RANK() OVER(PARTITION BY tag\_name ORDER BY COUNT(t.tag\_name) DESC) AS ranking

FROM users u

JOIN photos p on u.id = p.user\_id

JOIN photo\_tags pt on p.id = pt.photo\_id

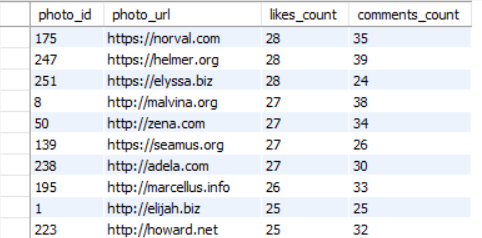
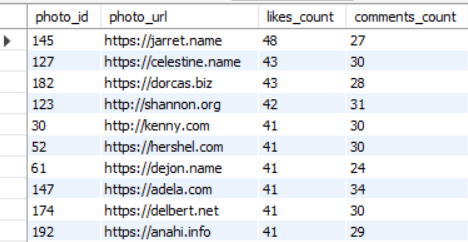
JOIN tags t on pt.tag\_id = t.id

GROUP BY 1, 2

) AS dt

WHERE ranking = 1;

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?**



We have data on photos only. So, we can check the correlation between user activities on photos.

The above table shows the 10 photos with highest likes and 10 photos with least likes. As we have ordered the table based on likes\_count column in descending order.

The above line chart shows the relation of user likes and comments. Ignoring few exceptions, we see that the posts which have higher number of likes also have higher number of comments.

The query used to get the output –

SELECT p.id AS photo\_id, p.image\_url AS photo\_url, COUNT(DISTINCT l.user\_id) AS likes\_count, COUNT(DISTINCT c.id) AS comments\_count

FROM photos p

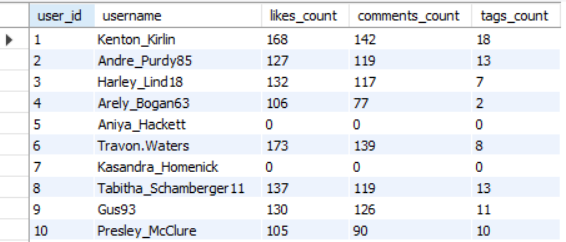
LEFT JOIN likes l on p.id = l.photo\_id

LEFT JOIN comments c on p.id = c.photo\_id

GROUP BY 1

ORDER BY 3 DESC;

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

****

The above table is a snippet of the output that we got. It shows the user id, user-name, number of likes, number of comments and number of tags each user have used.

The query used to get the output –

SELECT user\_id, username, SUM(likes\_count) AS likes\_count, SUM(comments\_count) AS comments\_count, SUM(tags\_count) AS tags\_count

FROM (

SELECT u.id AS user\_id, u.username, p.id AS photo\_id, COUNT(DISTINCT l.user\_id) AS likes\_count, COUNT(DISTINCT c.id) AS comments\_count, COUNT(DISTINCT tag\_id) AS tags\_count

FROM users u

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

LEFT JOIN likes l ON p.id = l.photo\_id

LEFT JOIN comments c ON p.id = c.photo\_id

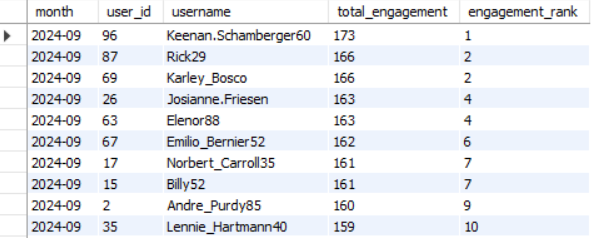
LEFT JOIN photo\_tags pt ON p.id = pt.photo\_id

GROUP BY 1, 2, 3

) dt

GROUP BY 1, 2;

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

****

The above table is a snippet of the output that we got. Its shows the month of year, user id, user-name, total engagement and engagement rank. The output is ordered by total engagement in descending order.

The query used to get the output –

SELECT DATE\_FORMAT(p.created\_dat, '%Y-%m') AS `month`,

u.id AS user\_id,

u.username,

(COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) AS total\_engagement,

RANK() OVER(PARTITION BY DATE\_FORMAT(p.created\_dat, '%Y-%m') ORDER BY (COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) DESC) AS engagement\_rank

FROM users u

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

LEFT JOIN likes l ON u.id = l.user\_id AND p.created\_dat = l.created\_at

LEFT JOIN comments c ON u.id = c.user\_id AND p.created\_dat = c.created\_at

WHERE DATE\_FORMAT(p.created\_dat, '%Y-%m') IS NOT NULL

GROUP BY 1, 2, 3;

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.**

****

The above table shows the hashtags that have been used in the post with highest average number of likes.

The query used to get the output –

WITH highest\_avg\_likes AS (

SELECT post\_id

FROM (

SELECT p.id AS post\_id, avg\_likes, RANK() OVER(ORDER BY avg\_likes DESC) AS ranking

FROM photos p

LEFT JOIN (

SELECT photo\_id, COUNT(user\_id) / (SELECT COUNT(DISTINCT user\_id) FROM photos) AS avg\_likes

FROM likes

GROUP BY 1

) l ON p.id = l.photo\_id

) dt

WHERE ranking = 1

)

SELECT al.post\_id, GROUP\_CONCAT(t.tag\_name) AS hashtags

FROM highest\_avg\_likes al

JOIN photo\_tags pt ON al.post\_id = pt.photo\_id

JOIN tags t ON pt.tag\_id = t.id

GROUP BY 1;

The query according to the question –

WITH tag\_likes AS (

SELECT t.id AS tag\_id,

tag\_name,

pt.photo\_id,

COUNT(DISTINCT l.user\_id) AS total\_likes,

AVG(COUNT(DISTINCT l.user\_id)) OVER(PARTITION BY t.id) AS avg\_likes

FROM tags t

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

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

GROUP BY 1, 2, 3

)

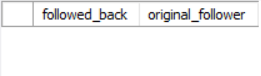
SELECT DISTINCT tag\_id, tag\_name

FROM tag\_likes

WHERE avg\_likes IN (SELECT MAX(avg\_likes) FROM tag\_likes)

ORDER BY 1;

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



We got an empty table for this because all the dates in the follows table are same, meaning everyone followed each other at the same time which is absurd.

The query used to get the output –

SELECT f1.follower\_id AS followed\_back, f1.followee\_id AS original\_follower

FROM follows f1

JOIN follows f2

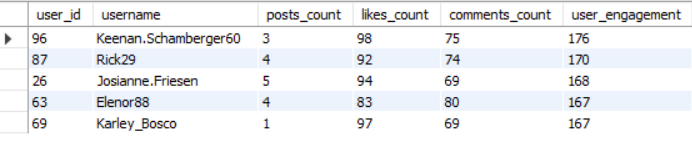
ON f1.follower\_id = f2.followee\_id

AND f1.followee\_id = f2.follower\_id

AND f1.created\_at > f2.created\_at;

**Subjective Questions**

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?**



Based on user engagement and activity levels, the users listed in the above table would be considered the most loyal or valuable users as they have the highest engagement among all the users.

A filter of posts greater than 0 is applied because the possibility of an account with 0 posts being a fake is high. So, those accounts are not considered.

To reward or incentivize loyal and valuable users:

1. **Exclusive Access:** Give these users early access to new features, products, or special content. This makes them feel like valued insiders with special privileges.
2. **Recognition:** Publicly acknowledge top users with leaderboards or a “Top Contributors” section. Use badges or titles to highlight their achievements and motivate them to maintain their status.
3. **Rewards:** Offer tangible rewards like gift cards, discounts, or free premium subscriptions. This provides a direct benefit for their continued loyalty and participation.
4. **Community Influence:** Let engaged users contribute by becoming moderators or beta testers, allowing them to have a say in shaping the platform and fostering deeper connection.
5. **Incentive Program:** Implement a system where users earn points or credits for their activity (likes, comments, posts) that can be redeemed for perks like merchandise or services, driving further engagement.

The query used to get the output is –

WITH cte AS (

SELECT u.id AS user\_id,

u.username,

COUNT(DISTINCT p.id) AS posts\_count,

COUNT(DISTINCT l.photo\_id) AS likes\_count,

COUNT(DISTINCT c.id) AS comments\_count,

COUNT(DISTINCT p.id) + COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id) AS user\_engagement,

DENSE\_RANK() OVER(ORDER BY COUNT(DISTINCT p.id) + COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id) DESC) AS drank

FROM users u

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

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

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

GROUP BY 1, 2

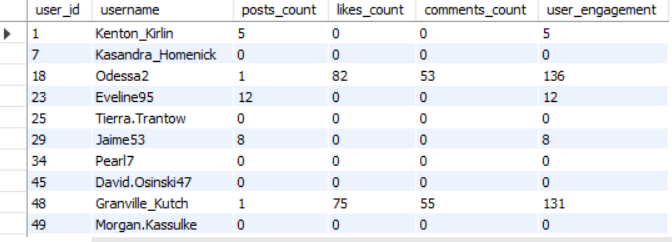
)

SELECT user\_id, username, posts\_count, likes\_count, comments\_count, user\_engagement

FROM cte

WHERE drank BETWEEN 1 AND 5 AND posts\_count > 0;

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



The above table is a snippet of the output we got. It shows the user id, user-name, number of posts, number of likes, number of comments and user engagement of the inactive users.

Strategies I would you recommend to re-engage them and encourage them to start posting or engaging again -

1. **Popular posts:** Show them posts which have got higher number of engagement and based on that he can see what kind of tags and photos to post.
2. **Personalized Posts:** Show them post based on their tags and most watch.
3. **Benefits of Engagement:** We can give them premium customers benefits of limited time if they post or increase their engagement to a certain level. That’ll surely motivate them. As who doesn’t want premium customer benefits for free.

The query used to get the output –

WITH cte AS (

SELECT u.id AS user\_id,

u.username,

COUNT(DISTINCT p.id) AS posts\_count,

COUNT(DISTINCT l.photo\_id) AS likes\_count,

COUNT(DISTINCT c.id) AS comments\_count,

COUNT(DISTINCT p.id) + COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id) AS user\_engagement,

DENSE\_RANK() OVER(ORDER BY COUNT(DISTINCT p.id) + COUNT(DISTINCT l.photo\_id) + COUNT(DISTINCT c.id)) AS drank

FROM users u

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

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

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

GROUP BY 1, 2

)

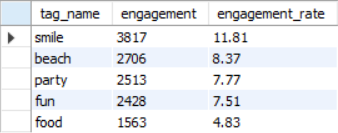
SELECT user\_id, username, posts\_count, likes\_count, comments\_count, user\_engagement

FROM cte

WHERE drank BETWEEN 1 AND 10

ORDER BY 1;

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

****

The above table shows the top 5 hashtags or content topics that have the highest engagement rates.

The highest engagement rates mean the posts with these hashtags are getting a higher number of likes and comments than the other post. So, we can make content and ads using these hashtags for delivering our message.

This information can guide content strategy and ad campaigns in following ways –

1. **Focusing on Popular Hashtags:** Use high-engagement tags like **smile** (11.81%), **beach** (8.37%), and **party** (7.77%) to reach a more active audience. These topics resonate well with users and should be central to content strategy.
2. **Creating Targeted Content:** Tailor content around the most engaging topics. For example, content related to **smile** or **fun** (7.51%) will likely attract more interaction, boosting user engagement.
3. **Optimizing Ads:** Incorporate high-engagement hashtags like **smile** and **party** into ad campaigns to increase relevance and attention. These tags have demonstrated strong user interest and can enhance ad performance.

The query used to get the output –

WITH count\_likes AS (

SELECT t.tag\_name, COUNT(l.user\_id) AS likes\_count

FROM tags t

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

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

GROUP BY 1

),

count\_posts AS (

SELECT t.tag\_name, COUNT(p.id) AS posts\_count

FROM tags t

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

LEFT JOIN photos p ON pt.photo\_id = p.id

GROUP BY 1

),

count\_comments AS (

SELECT t.tag\_name, COUNT(c.id) AS comments\_count

FROM tags t

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

LEFT JOIN comments c ON pt.photo\_id = c.photo\_id

GROUP BY 1

)

SELECT cl.tag\_name,

cl.likes\_count + cp.posts\_count + cc.comments\_count AS engagement,

ROUND((cl.likes\_count + cp.posts\_count + cc.comments\_count) \* 100 / SUM(cl.likes\_count + cp.posts\_count + cc.comments\_count) OVER(), 2) AS engagement\_rate

FROM count\_likes cl

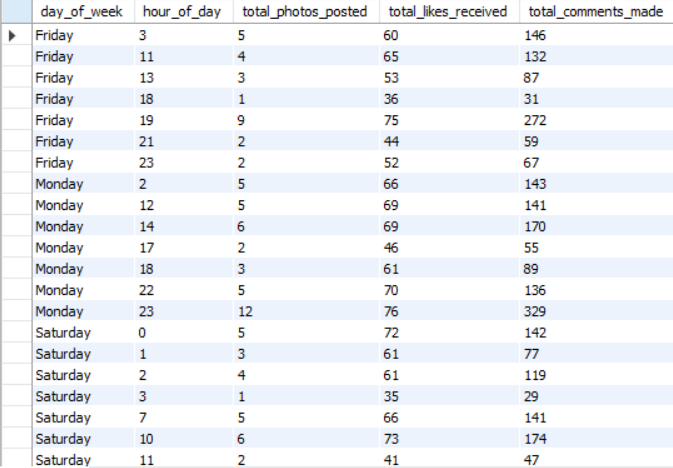
JOIN count\_posts cp ON cl.tag\_name = cp.tag\_name

JOIN count\_comments cc ON cl.tag\_name = cc.tag\_name

ORDER BY 2 DESC

LIMIT 5;

1. **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?**



The above table snippet shows user engagement patterns (photos posted, likes received, and comments made) by day of the week and hour of the day.

Key patterns and marketing insights based on the observation table –

1. **Peak Engagement Times:**

* Afternoon (11 AM to 12 PM) and evening (4 PM to 7 PM, 9 PM and 11 PM) are high engagement periods for photos, likes, and comments.

1. **Day of the Week Trends:**

* High activity on Tuesday and Thursday, with sustained engagement on Sunday and Monday. There is a dip on Friday and Wednesday.

1. **Late-Night Activity:**

* Strong engagement from 1 AM to 3 AM.

1. **Marketing Insights:**

* Schedule campaigns for peak times, particularly in the afternoon and evening on high-engagement days (Tuesday, Thursday, Sunday).
* Utilize late-night activity windows with promotions or interactive content, especially early in the week and late on Wednesday.
* Leverage weekend engagement (Saturday and Sunday) for brand awareness campaigns or targeted ads.

The query used to get the output –

SELECT

WEEKDAY(u.created\_at) AS day\_of\_week,

EXTRACT(HOUR FROM u.created\_at) AS hour\_of\_day,

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

COUNT(DISTINCT l.user\_id) AS total\_likes\_received,

COUNT(DISTINCT c.id) AS total\_comments\_made

FROM users u

LEFT JOIN photos p

ON u.id = p.user\_id

LEFT JOIN likes l

ON p.id = l.photo\_id

LEFT JOIN comments c

ON p.id = c.photo\_id

WHERE EXTRACT(HOUR FROM p.created\_dat) is not null

GROUP BY

day\_of\_week,

hour\_of\_day

ORDER BY

day\_of\_week,

hour\_of\_day;

****

The time of photo posted is same for all the photos. So, account created time is taken into consideration in the above query.

The query used to get this result is –

SELECT

WEEKDAY(p.created\_dat) AS day\_of\_week,

EXTRACT(HOUR FROM p.created\_dat) AS hour\_of\_day,

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

COUNT(DISTINCT l.user\_id) AS total\_likes\_received,

COUNT(DISTINCT c.id) AS total\_comments\_made

FROM photos p

LEFT JOIN likes l

ON p.id = l.photo\_id

LEFT JOIN comments c

ON p.id = c.photo\_id

GROUP BY

day\_of\_week,

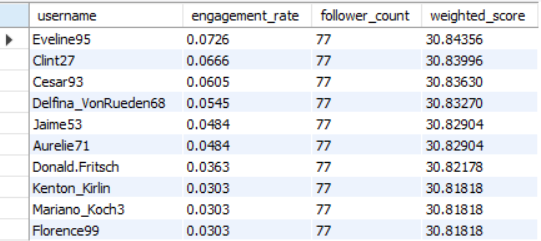
hour\_of\_day

ORDER BY

day\_of\_week,

hour\_of\_day;

1. **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?**

****

The above table shows the list of ideal candidates for influencer marketing campaign.

Based on user engagement rate and follower count, users who have a balance between engagement rate and follower count should be ideal candidates for influencer marketing campaigns.

Weighted Score is calculated adding 60 % engagement rate and 40 % followers.

To approach and collaborate with influencers, the following things can be done –

1. **Personalize Outreach**: Tailor messages by researching the influencer's content, highlighting shared values, and emphasizing why they’re a good fit for the campaign.
2. **Offer Mutual Benefits**: Provide clear compensation (monetary, free products, or exclusive access) and emphasize the potential for long-term partnerships.
3. **Encourage Creative Freedom**: Allow influencers to create authentic content that resonates with their audience while setting clear deliverables and expectations.
4. **Maintain Communication**: Keep open lines for feedback and collaboration throughout the campaign.
5. **Evaluate and Appreciate**: Track campaign success, share results with the influencer, and celebrate their contribution to foster lasting relationships.

The query used for the output –

WITH cte AS (

SELECT username, engagement\_rate, follower\_count,

(engagement\_rate \* 0.6 + follower\_count \* 0.4) AS weighted\_score

FROM (

SELECT u.id AS user\_id, u.username,

(COALESCE(posts\_count, 0) + COALESCE(likes\_count, 0) + COALESCE(comments\_count, 0)) \* 100 /

SUM((COALESCE(posts\_count, 0) + COALESCE(likes\_count, 0) + COALESCE(comments\_count, 0))) OVER() AS engagement\_rate,

follower\_count

FROM users u

LEFT JOIN (

SELECT user\_id, COUNT(\*) AS posts\_count

FROM photos

GROUP BY user\_id

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

LEFT JOIN (

SELECT user\_id, COUNT(DISTINCT photo\_id) AS likes\_count

FROM likes

GROUP BY user\_id

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

LEFT JOIN (

SELECT user\_id, COUNT(\*) AS comments\_count

FROM comments

GROUP BY user\_id

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

LEFT JOIN (

SELECT followee\_id, COUNT(DISTINCT follower\_id) AS follower\_count

FROM follows

GROUP BY followee\_id

) f2 ON u.id = f2.followee\_id

) dt

)

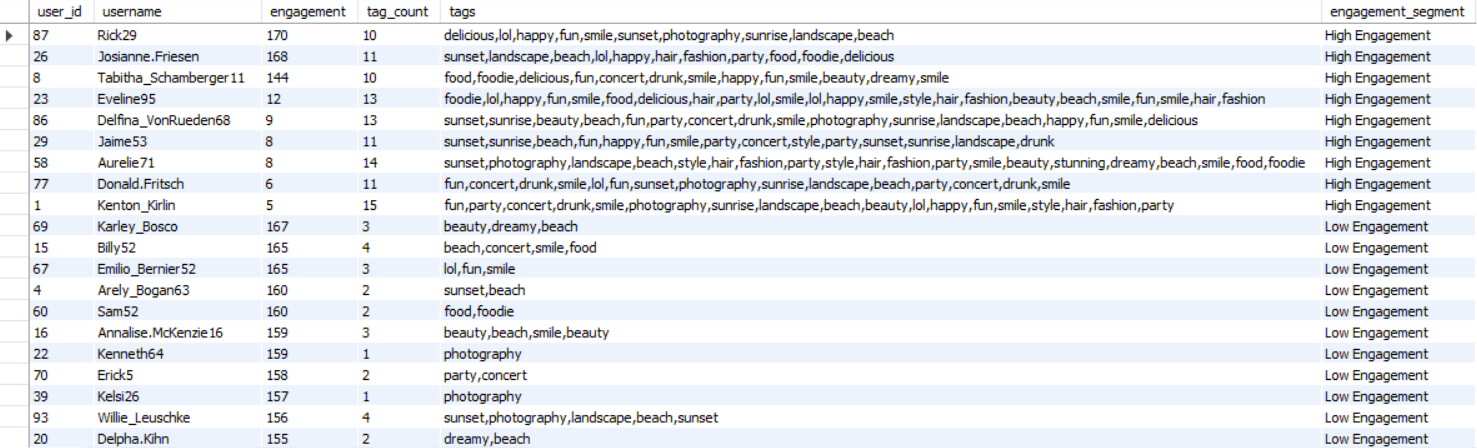
SELECT username, engagement\_rate, follower\_count, weighted\_score

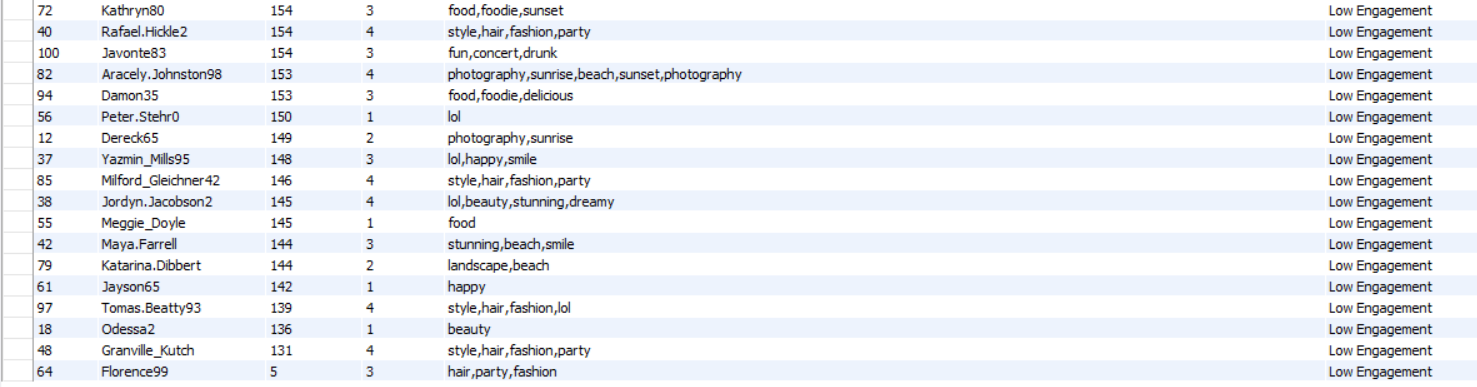
FROM cte

WHERE follower\_count = (select max(follower\_count) from cte) and engagement\_rate > 0

ORDER BY weighted\_score DESC;

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

****

****

Segmentation of users’ base for targeted marketing campaign or personalized recommendation –

1. **High Engagement Users:** Active users with high interaction and diverse tag usage, ideal for personalized recommendations, rewards, or influencer campaigns.
2. **Low Engagement Users:** Less active users with minimal interactions and fewer tags, requiring targeted re-engagement strategies or incentives.
3. **Tag-Based Segmentation**: Leverage users' tag preferences (e.g., food, photography) to create personalized campaigns or product recommendations, aligning content with user interests for better engagement.

**Targeting Strategy:** Offer exclusive access and rewards to high-engagement users while encouraging more activity from low-engagement users through personalized campaigns.

**Marketing Focus:** Use engagement and tag data to segment users, ensuring relevant content and offers for each group, maximizing overall engagement.

The query used to get the output –

WITH user\_engagement AS (

SELECT

u.id AS user\_id,

u.username,

COALESCE(p.engagement, 0) + COALESCE(l.engagement, 0) + COALESCE(c.engagement, 0) AS engagement,

COALESCE(t.tag\_count, 0) AS tag\_count

FROM

users u

LEFT JOIN (

SELECT user\_id, COUNT(DISTINCT id) AS engagement

FROM photos

GROUP BY user\_id

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

LEFT JOIN (

SELECT user\_id, COUNT(DISTINCT photo\_id) AS engagement

FROM likes

GROUP BY user\_id

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

LEFT JOIN (

SELECT user\_id, COUNT(DISTINCT id) AS engagement

FROM comments

GROUP BY user\_id

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

LEFT JOIN (

SELECT u.id AS user\_id,

COUNT(DISTINCT t.tag\_name) AS tag\_count

FROM

users u

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

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

LEFT JOIN tags t ON pt.tag\_id = t.id

GROUP BY u.id

) t ON u.id = t.user\_id

),

global\_max AS (

SELECT

MAX(engagement) AS max\_engagement,

MAX(tag\_count) AS max\_tag\_count

FROM user\_engagement

),

user\_tags AS (

SELECT

u.id AS user\_id,

group\_concat(t.tag\_name) AS tags

FROM

users u

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

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

LEFT JOIN tags t ON pt.tag\_id = t.id

GROUP BY u.id

),

user\_segments AS (

SELECT

e.user\_id,

e.username,

e.engagement,

e.tag\_count,

t.tags,

CASE

WHEN e.engagement < gm.max\_engagement / 3 AND e.tag\_count < gm.max\_tag\_count / 3 THEN 'Low Engagement'

WHEN e.engagement < 2 \* gm.max\_engagement / 3 AND e.tag\_count < 2 \* gm.max\_tag\_count / 3 THEN 'Moderate Engagement'

ELSE 'High Engagement'

END AS engagement\_segment

FROM user\_engagement e

LEFT JOIN user\_tags t ON e.user\_id = t.user\_id

CROSS JOIN global\_max gm -- Cross join to ensure you can use the global maximums

GROUP BY e.user\_id, e.username, e.engagement, e.tag\_count, t.tags, gm.max\_engagement, gm.max\_tag\_count

)

SELECT \*

FROM user\_segments

WHERE tag\_count > 0 AND tags IS NOT NULL and engagement\_segment in ('High Engagement', 'Low Engagement')

ORDER BY engagement\_segment, engagement DESC;

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

If data on ad campaigns (impressions, clicks, conversions) is available, then the key metrics to measure effectiveness and optimize future campaigns would be –

1. **Click-Through Rate (CTR)** = (Clicks / Impressions) \* 100

**Purpose**: This measures how well your ads are capturing attention. A higher CTR indicates that more people who see your ad are clicking on it, signaling the ad’s relevance and appeal.

1. **Conversion Rate (CVR)** = (Conversions / Clicks) \* 100

**Purpose**: CVR tells you how effective your ad is at driving desired actions (e.g., sales, sign-ups) after users click. It helps assess the quality of the traffic generated by the ad.

1. **Cost per Acquisition (CPA)** = Total Spend / Conversions

**Purpose**: CPA measures how much you are spending to acquire one customer or conversion. It’s essential for understanding the cost-effectiveness of your campaigns, especially for budget allocation.

1. **Return on Ad Spend (ROAS)** = Revenue / Total Spend

**Purpose**: ROAS evaluates the overall financial return of your campaign. A higher ROAS means you’re earning more revenue compared to what you're spending on ads, making it a critical metric for determining profitability.

For optimization of future campaign –

1. **Allocate budget to high performing segments** **-** Focus your spending on users that show the best CTR, CVR, and ROAS. By concentrating resources on what works, you maximize return and reduce wasted spend.
2. **Adjust bids for better performing ads -** Increase bids for ads or keywords that generate strong engagement and conversions to capture more traffic. Lower bids or pause underperforming ads to optimize spending efficiency.
3. **Retargeting users who didn’t convert -** Implement retargeting strategies to show ads to users who clicked but didn’t convert. By re-engaging potential customers, you can increase conversion rates without having to find new audiences.
4. **Optimizing landing pages to improve conversion -** Ensure that the landing pages users reach after clicking on an ad are relevant, fast-loading, and have a clear call to action. By improving the user experience, you can boost conversion rates and overall campaign performance.
5. **How can you use user activity data to identify potential brand ambassadors or advocates who could help promote Instagram's initiatives or events?**

**High Follower Count and High Engagement**:

* 1. Users with a high number of followers and consistently high numbers of likes and comments are prime candidates.
  2. These users not only have a large reach but also have an engaged audience.

**Consistency in Engagement**:

* 1. Users who consistently receive high numbers of likes and comments across multiple posts may indicate genuine interest and influence, as opposed to one-time viral activity.

**Identify Users with Balanced Engagement**:

* 1. We should focus on users who both generate and receive engagement, as these users are more likely to influence their network and drive engagement for Instagram initiatives.

WITH followers AS(

SELECT u.id user\_id,

username,

COUNT(follower\_id) AS follower\_count

FROM users u

LEFT JOIN follows f ON u.id=f.followee\_id

GROUP BY 1,2

),

engagement AS(

SELECT u.id AS user\_id,

username,

p.id AS photo\_id,

COUNT(DISTINCT c.user\_id) comments,

COUNT(DISTINCT l.user\_id) likes

FROM users u

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

LEFT JOIN comments c ON c.photo\_id=p.id

LEFT JOIN likes l ON l.photo\_id=p.id

GROUP BY 1,2,3

)

SELECT f.user\_id,

f.username,

follower\_count,

SUM(comments) + SUM(likes) total\_likes

FROM followers f

JOIN engagement e ON f.user\_id=e.user\_id

WHERE follower\_count = (SELECT MAX(follower\_count) FROM followers)

GROUP BY 1,2,3

ORDER BY 4 DESC LIMIT 4;

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

**Problem Statement -** You are hired as a data analyst at Meta and asked to collaborate with Marketing team. Marketing teams wants to leverage Instagram's user data to develop targeted marketing strategies that will increase user engagement, retention, and acquisition. Provide insights and recommendations to address the following objectives.

If the objective and subjective question were not given, then key metrics to focus –

1. **Followers & Following**:

* **Calculate Total Followers**: Assess the influence and reach of users.
* **Calculate Total Following**: Understand users’ engagement levels and content consumption patterns.

1. **Posts**:

* **Count Total Posts**: Track overall activity levels of users.
* **Categorize Posting Frequency**:
  + **High Activity**: Users who post frequently.
  + **Moderate Activity**: Users who post occasionally.
  + **Low Activity**: Users who post rarely.
  + **Non-Posters**: Users who haven’t posted at all.
* **Engagement per Post**: Analyze likes and comments per post to determine content effectiveness.

1. **Engagement (Likes & Comments)**:

* **Total Likes**: Measure overall content interaction and popularity.
* **Total Comments**: Evaluate user discussions and engagement levels.
* **Engagement Rate**: Calculate the ratio of likes and comments to the number of followers to assess content resonance.
* **Identify Active Engagers**: Recognize users who consistently engage with content and encourage them to share more.
* **No Engagement**: Identify users who have not engaged (liked or commented) on any content. Target these users with personalized content or prompts to encourage interaction.

1. **Content Types (Hashtags)**:

* **Analyze Hashtag Usage**: Track which hashtags users frequently employ to enhance visibility.
* **Engagement per Hashtag**: Measure likes and comments associated with specific hashtags to identify trending topics.
* **Encourage Hashtag Participation**: Motivate users to use popular or trending hashtags to broaden their reach.

For increasing user engagement, retention and acquisition –

1. **Implement a Reward System**:

* Create incentives where users receive premium features for free for a limited time when their posts reach a certain number of likes or views. This can motivate users to create more engaging content.

1. **Personalized Content Delivery**:

* Send tailored posts and recommendations based on user preferences and past interactions. This encourages users to engage more with the content they find relevant.

1. **Influencer Partnerships and Ads**:

* Collaborate with influencers to promote the platform, tapping into their follower base for wider reach. Additionally, invest in targeted ads across various platforms to attract new users.

1. **Referral Rewards**:

* Offer bonuses or rewards for users who invite their friends to join the platform. This not only boosts user acquisition but also fosters community growth.

1. **New User Rewards**:

* Provide special rewards for new users, such as bonus features or discounts, to encourage them to engage with the platform from the start.

1. **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**

To do this we’ll use Data Manipulation Language function ‘UPDATE’.

The QUERY to do this will be –

UPDATE User\_Interactions

SET Engagement\_Type = ‘Heart’

WHERE Engagement\_Type = ‘Like’;

UPDATE function will change all the instances of like to heart to make it familiar with Instagram terminology.

In SQL, the **UPDATE statement** is used to modify existing records in a table. Whether you need to update a single record or multiple rows at once, SQL provides the necessary functionality to make these changes. The UPDATE statement is essential for maintaining the integrity of your data by allowing you to correct or **modify existing** entries in your database.