**Learners have to come up with a Report to support the answers to the following questions and suggestions**

**Objective Questions**

1. **Are there any tables with duplicate or missing null values? If so, how would you handle them?**
   * There are no duplicate or missing null values in the given Data Set.
   * If it’s there, I would use the **COALESCE** function or **ISNULL** to replace null values.
   * First most thing will ask the client for default values to replace the null values.
   * If the Numerical column which contains the NULL value, then will replace null values with mostly mean, median or mode.
   * If any text column contains the Null value, then will replace with “Not Applicable” or “NA” any default word.
2. **What is the distribution of user activity levels (e.g., number of posts, likes, comments) across the user base?**

**SQL Query:**

with cte as(

select u.id, u.username,

coalesce(post\_count,0) as post\_count,

coalesce(like\_count,0) as like\_count,

coalesce(comment\_count,0) as comment\_count,

(coalesce(post\_count,0) + coalesce(like\_count,0) + coalesce(comment\_count,0)) as total\_activity,

case

when (coalesce(post\_count,0) + coalesce(like\_count,0) + coalesce(comment\_count,0)) >165 then "Highly activity"

when (coalesce(post\_count,0) + coalesce(like\_count,0) + coalesce(comment\_count,0)) between 100 and 165 then "Moderately active"

when (coalesce(post\_count,0) + coalesce(like\_count,0) + coalesce(comment\_count,0)) < 100 then "Less activity"

else "Inactive"

end as user\_activity

from users u

left join

(

select user\_id, count(\*) as comment\_count

from comments

group by user\_id

) c

on u.id = c.user\_id

left join

( select user\_id, count(\*) as post\_count

from photos

group by user\_id

) p

on u.id= p.user\_id

left join

(select user\_id, count(\*) as like\_count

from likes

group by user\_id

) l

on u.id = l.user\_id)

select user\_activity, count(\*) as count from cte group by user\_activity;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.  
  
Observation:**

The distribution of user activity levels among the users :

* The moderately active users are more when compared to other users. 59% of users are moderately active.
* The Highly active users are 18% among the users. These users are more interactive on Instagram regarding posts, likes and comments.
* The Less active users are 10% among the users. These users are less interactive compared to other users.
* The Inactive users are 13% among all the users. These users are inactive state, they have the account, but they are not in actively participating in Instagram.

**Approach:**

I categorized the user activity based on the total activity, if the total activity is greater than 165 then it is “Highly active”, if it is between 100 and 165 then “Moderately active” and if it is less than 100 then it is “Less active”.

The total activity comes from the sum of the user's posts, user likes, and user comments.

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

**SQL QUERY:**

select round(avg(tagcount),2) as avgpostcount from (

select photo\_id, count(\*) as tagcount

from photo\_tags

group by photo\_id)as t1;

**Output :**

**A close up of a number

AI-generated content may be incorrect.**

The average number of tags per post is **2.64.**

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

**SQL QUERY:**

with cte1 as (

select u.id, u.username,

coalesce(like\_count,0) as like\_count,

coalesce(comment\_count,0) as comment\_count,

(coalesce(like\_count,0) + coalesce(comment\_count,0)) as total\_activity

from users u

left join

(

select user\_id, count(\*) as comment\_count

from comments

group by user\_id

) c

on u.id = c.user\_id

left join

(select user\_id, count(\*) as like\_count

from likes

group by user\_id

) l

on u.id = l.user\_id)

select \*, dense\_rank() over(order by total\_activity desc) as user\_rank from cte1;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | username | like\_count | comment\_count | total\_activity | users\_rank |
| 71 | Nia\_Haag | 257 | 257 | 514 | 1 |
| 5 | Aniya\_Hackett | 257 | 257 | 514 | 1 |
| 41 | Mckenna17 | 257 | 257 | 514 | 1 |
| 21 | Rocio33 | 257 | 257 | 514 | 1 |
| 66 | Mike.Auer39 | 257 | 257 | 514 | 1 |
| 54 | Duane60 | 257 | 257 | 514 | 1 |
| 24 | Maxwell.Halvorson | 257 | 257 | 514 | 1 |
| 14 | Jaclyn81 | 257 | 257 | 514 | 1 |
| 57 | Julien\_Schmidt | 257 | 257 | 514 | 1 |
| 91 | Bethany20 | 257 | 257 | 514 | 1 |
| 76 | Janelle.Nikolaus81 | 257 | 257 | 514 | 1 |
| 75 | Leslie67 | 257 | 257 | 514 | 1 |
| 36 | Ollie\_Ledner37 | 257 | 257 | 514 | 1 |

I used dense\_rank() function to rank them based on the total\_activity.   
  
**Observation:**The above-mentioned users are the top users on the bases of highest engagement on likes and comments. Every user has highest no. of likes and comments on total posts.

**Approach:**Dense rank will rank the users based on the given condition, but it not skips the rank which are after similar positions.I used dense rank() function to rank them based on the total activity, I summed up the total likes, total comments of the user to calculate the total activity. Later I ranked on total activity in descending order.

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

**SQL Query:**

select u.id, u.username, coalesce(total\_followers,0) as total\_followers , coalesce(total\_following,0) as total\_following

from users u

left join

(

select followee\_id, count(follower\_id) as total\_followers

from follows

group by followee\_id

) f

on u.id = f.followee\_id

left join

(

select follower\_id,count(followee\_id )as total\_following

from

follows

group by follower\_id

) f1

on u.id = f1.follower\_id

order by total\_following desc, total\_followers desc;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **username** | **followers** | **following** |
| 100 | Javonte83 | 76 | 99 |
| 2 | Andre\_Purdy85 | 76 | 99 |
| 3 | Harley\_Lind18 | 76 | 99 |
| 4 | Arely\_Bogan63 | 76 | 99 |
| 5 | Aniya\_Hackett | 76 | 99 |
| 6 | Travon.Waters | 76 | 99 |
| 12 | Dereck65 | 76 | 99 |
| 8 | Tabitha\_Schamberger11 | 76 | 99 |
| 9 | Gus93 | 76 | 99 |
| 10 | Presley\_McClure | 76 | 99 |

These are the list of only top 10 records.

**Observation:**The above-mentioned users are having the highest number of followers and following on the Instagram.

**Approach:**I had taken the followee id and follower id of each user and used count function to count the followers and following of the users.

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

**SQL Query:**

select u.id, u.username,

coalesce(sum(totaleng),0) / count(p.id) as avg\_eng\_rate

from users u

left join

(

select p.user\_id, p.id as post\_id, coalesce(total\_likes,0) + coalesce(total\_comments,0) as totaleng

from photos p

left join

(

select photo\_id, count(user\_id) as total\_likes

from likes

group by photo\_id

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

left join

(

select photo\_id, count(id) as total\_comments

from comments

group by photo\_id

) c on p.id = c.photo\_id

) e on u.id = e.user\_id

left join

photos p

on u.id = p.user\_id

group by u.id,u.username

order by avg\_eng\_rate desc

limit 10;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |
| --- | --- | --- |
| **id** | **username** | **avg\_eng\_rate** |
| 55 | Meggie\_Doyle | 75 |
| 73 | Jaylan.Lakin | 73 |
| 48 | Granville\_Kutch | 71 |
| 22 | Kenneth64 | 70 |
| 69 | Karley\_Bosco | 68 |
| 94 | Damon35 | 68 |
| 87 | Rick29 | 68 |
| 18 | Odessa2 | 67 |
| 43 | Janet.Armstrong | 66.8 |
| 52 | Zack\_Kemmer93 | 66.6 |

These are the list of only top 10 records

**Observation:**The above-mentioned data is the average engagement rate based on likes and comments per post of each user. The highest average engagement rate is 75 for the user Meggie Doyle.

**Approach:**I summed up the likes and comments of the each post of the user and divided by total counts of the posts of the user to achieve the average engagement rate.

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

**SQL QUERY:**

select id, username from users

where id not in (select user\_id from likes) ;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |
| --- | --- |
| id | username |
| 1 | Kenton\_Kirlin |
| 7 | Kasandra\_Homenick |
| 23 | Eveline95 |
| 25 | Tierra.Trantow |
| 29 | Jaime53 |
| 34 | Pearl7 |
| 45 | David.Osinski47 |
| 49 | Morgan.Kassulke |
| 51 | Mariano\_Koch3 |
| 53 | Linnea59 |
| 58 | Aurelie71 |
| 59 | Cesar93 |
| 64 | Florence99 |
| 68 | Franco\_Keebler64 |
| 74 | Hulda.Macejkovic |
| 77 | Donald.Fritsch |
| 80 | Darby\_Herzog |
| 81 | Esther.Zulauf61 |
| 83 | Bartholome.Bernhard |
| 86 | Delfina\_VonRueden68 |
| 88 | Clint27 |
| 89 | Jessyca\_West |
| 90 | Esmeralda.Mraz57 |

This is the list of users who never liked the posts. There are mostly in inactive state.   
  
**Observation:**The above-mentioned users are inactive on Instagram; they never ever liked any posts on the Instagram.   
These users need to be targeted to make them active on Instagram to make them posts the photos, like and comments the posts of the other users.   
This can be achieved by giving them any reward or badges, by campaigning events etc.,

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

**SQL QUERY:**

SELECT

p.id AS post\_id,

p.user\_id,

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

COUNT(DISTINCT c.user\_id) AS total\_comments

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

p.id, p.user\_id

ORDER BY

(COUNT(DISTINCT l.user\_id) + COUNT(DISTINCT c.user\_id)) DESC

LIMIT 10;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |  |
| --- | --- | --- | --- |
| post\_id | user\_id | total\_likes | total\_comments |
| 13 | 3 | 40 | 39 |
| 118 | 43 | 39 | 35 |
| 123 | 44 | 42 | 31 |
| 127 | 46 | 43 | 30 |
| 145 | 52 | 48 | 27 |
| 147 | 55 | 41 | 34 |
| 157 | 59 | 33 | 39 |
| 196 | 73 | 38 | 35 |
| 227 | 87 | 39 | 35 |
| 225 | 87 | 37 | 36 |

For the photo\_id’s mentioned in the above table got the likes and comments more. So for the specific posts which got likes and comments more can be used to engage ads.

**SQL QUERY:**SELECT u.id AS user\_id, u.username, t.tag\_name, count(t.tag\_name) as count  
FROM users  
JOIN photos p ON u.id = p.user\_id  
JOIN photo\_tags pt ON p.id = pt.photo\_idJOIN tags t ON pt.tag\_id = t.idgroup by u.id, u.username, t.tag\_namehaving count(t.tag\_name) >2ORDER BY  
 u.id, t.tag\_name;  
  
**Output:  
A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |
| --- | --- | --- |
| **username** | **tagname** | **count** |
| Clint27 | beach | 6 |
| Eveline95 | smile | 5 |
| Clint27 | dreamy | 4 |
| Andre\_Purdy85 | fun | 3 |
| Andre\_Purdy85 | smile | 3 |
| Tabitha\_Schamberger11 | smile | 3 |
| Gus93 | stunning | 3 |
| Eveline95 | hair | 3 |
| Eveline95 | lol | 3 |
| Cesar93 | fun | 3 |
| Adelle96 | smile | 3 |
| Clint27 | sunrise | 3 |
| Clint27 | sunset | 3 |

For the above users by the help of tags we can personalise the ads based on the tags they are using.

**SQL QUERY:**select t.tag\_name, count(\*) AS usage\_count

from photo\_tags pt

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

join photos p on pt.photo\_id = p.id

group by t.tag\_name

order by count(\*) desc

limit 10;

**OUTPUT:  
A screenshot of a computer

AI-generated content may be incorrect.**

These are the tag\_names which are used most on the posts, by using the tag\_names we can personalize the ads and target the users.

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?**

**SQL Query:**  
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 p.id  
ORDER BY likes\_count DESC;

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Observation:**We have data on photos only. So, we can check the correlation between user activities on photos. Like no. of likes and no. of comments of each post. We can get the data like highest likes of the post and the highest comments of the post.   
  
The above line chart shows the correlation between user likes and comments. Ignoring a few exceptions, we see that the posts that have a higher number of likes also have a higher number of comments.   
  
 **There is no “content types” column in any Table.**   
If there are any column “content types” then I will sum the total likes, total comments, and average of watch time group by “content types” Then I will get all the required data related to all content types like photos, videos, and reels. Then I can correlate the user activity levels and specific content types.

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

**SQL QUERY:**with total as   
(  
select u.username,   
coalesce(like\_count,0) as total\_likes,  
coalesce(comment\_count,0) as total\_comments,  
coalesce(tag\_count,0) as total\_tags  
from users u   
left join (  
select p.user\_id, count(l.user\_id) as like\_count  
from likes l   
join   
photos p on l.user\_id= p.user\_id   
group by p.user\_id  
) like\_cont   
on u.id = like\_cont.user\_id   
left join   
(  
select p.user\_id, count( c.user\_id) as comment\_count  
from comments c   
join   
photos p   
on c.user\_id = p.user\_id   
group by p.user\_id  
) comment\_cont   
on u.id = comment\_cont.user\_id  
left join   
(  
SELECT p.user\_id, COUNT(pt.tag\_id) AS tag\_count  
from photo\_tags pt join photos p  
on pt.photo\_id = p.id  
group by p.user\_id  
) phototagcount   
on   
u.id = phototagcount.user\_id  
order by u.id  
)  
  
select \* from total order by username;  
  
**Output:  
A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |  |
| --- | --- | --- | --- |
| **username** | **total\_likes** | **total\_comments** | **total\_tags** |
| Adelle96 | 480 | 300 | 15 |
| Aiyana\_Hoeger | 88 | 66 | 5 |
| Alek\_Watsica | 222 | 204 | 5 |
| Alexandro35 | 465 | 290 | 7 |
| Alysa22 | 150 | 152 | 6 |

These are only list of 5 records.   
  
**Observation:**The above-mentioned output are the list of users and their total likes, total comment and total tags.  
This data can be further used to personalize the Ads for the particular user based on their likes and activity.   
For example, user is liking post of gaming and using tags of it, then we can push the Ads of Gaming consoles, New Games, PlayStation etc., to that user.

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

**SQL QUERY:**with total as

(

select u.username,

coalesce(like\_count,0) as total\_likes,

coalesce(comment\_count,0) as total\_comments,

coalesce(tag\_count,0) as total\_tags

from users u

left join (

select p.user\_id, count(l.user\_id) as like\_count

from likes l

join

photos p on l.user\_id= p.user\_id

group by p.user\_id

) like\_cont

on u.id = like\_cont.user\_id

left join

(

select p.user\_id, count( c.user\_id) as comment\_count

from comments c

join

photos p

on c.user\_id = p.user\_id

group by p.user\_id

) comment\_cont

on u.id = comment\_cont.user\_id

left join

(

SELECT p.user\_id, COUNT(pt.tag\_id) AS tag\_count

from photo\_tags pt join photos p

on pt.photo\_id = p.id

group by p.user\_id

) phototagcount

on

u.id = phototagcount.user\_id

order by u.id

)

select \* , (total\_likes + total\_comments + total\_tags) as sum,

dense\_rank() over(order by (total\_likes + total\_comments + total\_tags) desc) as "rank"

from total order by "rank";

**Output:  
A screenshot of a computer

AI-generated content may be incorrect.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **username** | **total\_likes** | **total\_comments** | **total\_tags** | **total\_activity** | **rank** |
| Josianne.Friesen | 470 | 345 | 11 | 826 | 1 |
| Adelle96 | 480 | 300 | 15 | 795 | 2 |
| Janet.Armstrong | 430 | 360 | 5 | 795 | 2 |
| Alexandro35 | 465 | 290 | 7 | 762 | 3 |
| Kathryn80 | 425 | 320 | 3 | 748 | 4 |

These are the only list of 5 records.

I ranked the users based on the total\_activity. I used dense\_rank() function to rank.

The total activity is sum of the total\_likes, total\_comments and total\_tags.  
**Observation:**The above-mentioned users are the top users on the bases of highest total engagement(total likes, total comments, total tags).

**Approach:**Dense rank will rank the users based on the given condition, but it not skips the rank which are after similar positions.I used dense rank() function to rank them based on the total activity, I summed up the total likes, total comments and total tags of the user to calculate the total activity. Later I ranked on total activity in descending order.

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.  
     
   SQL QUERY:**with cte as(  
   select tag\_name, round(avg(like\_count),2) as avg\_likes  
   from   
   tags t   
   join  
   photo\_tags pt on t.id= pt.tag\_id  
   join  
   photos p   
   on pt.photo\_id = p.id   
   left join  
   (   
   select photo\_id, count(user\_id) as like\_count   
   from likes   
   group by photo\_id   
   ) l  
   on l.photo\_id = pt.photo\_id   
   group by tag\_name  
   )

select \* from cte where avg\_likes = (select max(avg\_likes) from cte) ;  
  
**Output:**A screenshot of a computer

AI-generated content may be incorrect.  
**Observation:**  
For the Hashtag name **“Dreamy”** has the highest average likes of **35.75**

1. **Retrieve the users who have started following someone after being followed by that person.  
     
   SQL QUERY:**select f1.follower\_id as user\_id, f1.followee\_id AS followed\_user\_id  
   from follows f1  
   join follows f2  
   on f1.follower\_id = f2.followee\_id  
   and f1.followee\_id = f2.follower\_id  
   where f1.created\_at > f2.created\_at  
   order by f1.follower\_id, f1.followee\_id; **OUTPUT:**

For the above question, I had written the query but, in the data, the created date consists same for all. The data set that they provided has the same date for created\_date.

**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?**

**SQL QUERY:**with total as   
(  
select u.username,   
coalesce(like\_count,0) as total\_likes,  
coalesce(comment\_count,0) as total\_comments,  
coalesce(tag\_count,0) as total\_tags  
from users u   
left join (  
select p.user\_id, count(l.user\_id) as like\_count  
from likes l   
join   
photos p on l.user\_id= p.user\_id   
group by p.user\_id  
) like\_cont   
on u.id = like\_cont.user\_id   
left join   
(  
select p.user\_id, count( c.user\_id) as comment\_count  
from comments c   
join   
photos p   
on c.user\_id = p.user\_id   
group by p.user\_id  
) comment\_cont   
on u.id = comment\_cont.user\_id  
left join   
(  
SELECT p.user\_id, COUNT(pt.tag\_id) AS tag\_count  
from photo\_tags pt join photos p  
on pt.photo\_id = p.id  
group by p.user\_id  
) phototagcount   
on   
u.id = phototagcount.user\_id  
order by u.id  
)  
select \* , (total\_likes + total\_comments + total\_tags) as sum,  
dense\_rank() over(order by (total\_likes + total\_comments + total\_tags) desc) as "rank"  
from total order by "rank";  
**OUTPUT:  
A screenshot of a table

AI-generated content may be incorrect.  
  
Insight**: Based on user engagement and activity levels, I consider these users are the most loyal or valuable. I ranked based on the

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| username | total\_likes | total\_comments | total\_tags | sum | rank |
| Josianne.Friesen | 470 | 345 | 11 | 826 | 1 |
| Adelle96 | 480 | 300 | 15 | 795 | 2 |
| Janet.Armstrong | 430 | 360 | 5 | 795 | 2 |
| Alexandro35 | 465 | 290 | 7 | 762 | 3 |
| Kathryn80 | 425 | 320 | 3 | 748 | 4 |
| Travon.Waters | 410 | 310 | 8 | 728 | 5 |

**Recommendation:**   
-> I would reward the top users with the badges like “Top contributors” and “Super Influencer”   
-> Providing early access to new features.  
-> Display ranking on a Leaderboard.   
-> Giving loyal users access to exclusive communities.  
-> Provide monetary bonuses for top performers.  
-> Giving offers In-App purchases.  
-> Allow influential users to earn money through referrals.

1. **For inactive users, what strategies would you recommend to re-engage them and encourage them to start posting or engaging again?  
     
   SQL QUERY:**with total as   
   (  
   select u.username,   
   coalesce(like\_count,0) as total\_likes,  
   coalesce(comment\_count,0) as total\_comments,  
   coalesce(tag\_count,0) as total\_tags  
   from users u   
   left join (  
   select p.user\_id, count(l.user\_id) as like\_count  
   from likes l   
   join   
   photos p on l.user\_id= p.user\_id   
   group by p.user\_id  
   ) like\_cont   
   on u.id = like\_cont.user\_id   
   left join   
   (  
   select p.user\_id, count( c.user\_id) as comment\_count  
   from comments c   
   join   
   photos p   
   on c.user\_id = p.user\_id   
   group by p.user\_id  
   ) comment\_cont   
   on u.id = comment\_cont.user\_id  
   left join   
   (  
   SELECT p.user\_id, COUNT(pt.tag\_id) AS tag\_count  
   from photo\_tags pt join photos p  
   on pt.photo\_id = p.id  
   group by p.user\_id  
   ) phototagcount   
   on   
   u.id = phototagcount.user\_id  
   order by u.id  
   )  
   select \* , (total\_likes + total\_comments + total\_tags) as sum,  
   dense\_rank() over(order by (total\_likes + total\_comments + total\_tags) desc) as "rank"  
    from total  
    order by 'rank' desc;   
     
     
     
     
     
     
     
     
     
     
     
   **OUTPUT:**A screenshot of a computer

   AI-generated content may be incorrect.

**Insight:** These are the users who are not active. I listed out only five.

**Recommendation**:

* + Reward users for returning and engaging. By assigning Come back and earn a “Welcome Back Badge**”.**
  + Giving points to users whenever they interact with posts. Later, that points can be redeemed in App-Purchases.
  + Rewarding the users on referral or inviting new users.
  + Show only relevant posts or videos based on the previous information.
  + Creating a campaign for the user like “7-days posting Challenge” and get rewards after completing it.
  + Getting feedback from them to know the reason and implementing it in the next updates.

1. **Which hashtags or content topics have the highest engagement rates? How can this information guide content strategy and ad campaigns?  
     
   SQL QUERY:**select t.tag\_name, count(tag\_id) as hashtag\_count  
   from   
   photo\_tags pt  
   join  
   tags t on pt.tag\_id = t.id  
   group by  
   tag\_name;  
     
   **OUTPUT:**A screenshot of a computer

   AI-generated content may be incorrect.

**Insight:** These are the top 5 Hashtags are used in the posts.

**Recommendation:**

* + This information can be helpful for the content strategy and ad campaign. It encourages users to use trending hashtags in their posts.
  + The content creator can concentrate more on the hashtags are trending and create content around it.
  + Influencers can collaborate with others which are top – performing in the categories.
  + Feature highly engaged user posts to increase visibility.
  + With the help of Hashtags, we can target the specific users for Ads.
  + For Example, the users who are used #Gaming tags, for that users we can promote Ads which are gaming related like gaming headsets, new games, gaming merchandise etc.,
  + Promoting sponsored content under trending hashtags.

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?  
     
   SQL QUERY:**SELECT  
   DAYNAME(u.created\_at) AS day\_of\_week,  
   HOUR (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 HOUR (p.created\_dat) is not null   
   GROUP BY  
   day\_of\_week, hour\_of\_day  
   ORDER BY  
   day\_of\_week, hour\_of\_day; **Output:  
   A screenshot of a computer

   AI-generated content may be incorrect.**

* **Insight:**    
  The dataset provides essential user demographic information, enabling the creation of tailored content and targeted campaigns.  
  The posting time for all posts appears the same in the dataset, suggesting an opportunity to optimize scheduling once actual user activity patterns are understood.  
    
  **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.

* **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.

* **Late-Night Activity:**

Strong engagement from 1 AM to 3 AM.

* **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.

**Recommendations:**

**Leverage User Activity Timings**  
Identifying when users are most active (e.g., 2 PM to 4 PM) helps you schedule content or ads for maximum visibility.

If analytics show higher engagement in the early afternoon, push marketing messages and sponsored posts during that window to increase click-through rates and conversions.

**Personalize Content by Age Group  
Younger Users (Under 24)**Preferred Format: Short-form videos like Reels.Reason: Quick, visually appealing content tends to resonate with Gen Z’s faster consumption habits. **Older Users (25 to 40)**Preferred Format: Longer, more informative videos.Reason: This age group often values in-depth information and detailed insights, making longer content more engaging. **Post During Peak Engagement Hours**Timing the content when user engagement is highest increases the likelihood of interaction (likes, comments, shares).Use data to identify and confirm the time slots where your audience is most active, then schedule or boost ads accordingly. **Customize Ad Creatives by Gender & Location**Different genders and regions may have distinct cultural preferences and purchasing behaviors.Tailor ad visuals, language, and offers to match local tastes or gender-specific interests, leading to higher relevance and engagement. **Collaborate with Influencers from High-Engagement Segments**Influencers who align with your target demographics (e.g., specific age groups or locations) can amplify your reach and credibility.Identify and partner with content creators who consistently achieve strong engagement among your desired audience, and incorporate them into your marketing campaigns.

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?  
     
   SQL QUERY:**WITH UserEngagement AS (  
   SELECT   
   u.id AS user\_id,  
   u.username,  
   COUNT(DISTINCT f.follower\_id) AS total\_followers,  
   COUNT(DISTINCT p.id) AS total\_posts,  
   COALESCE(AVG(like\_count), 0) AS avg\_likes,  
   COALESCE(AVG(comment\_count), 0) AS avg\_comments,  
   (COALESCE(AVG(like\_count), 0) + COALESCE(AVG(comment\_count), 0) ) AS avg\_engagement\_rate  
   FROM users u  
   LEFT JOIN follows f ON u.id = f.followee\_id -- Count followers  
   LEFT JOIN photos p ON u.id = p.user\_id -- Count posts  
   LEFT JOIN (  
   SELECT photo\_id, COUNT(user\_id) AS like\_count FROM likes GROUP BY photo\_id  
   ) l ON p.id = l.photo\_id  
   LEFT JOIN (  
   SELECT photo\_id, COUNT(id) AS comment\_count FROM comments GROUP BY photo\_id  
    ) c ON p.id = c.photo\_id  
    GROUP BY u.id, u.username  
   )  
   SELECT   
   user\_id,  
   username,  
   total\_followers,   
   total\_posts,  
   avg\_likes,   
   avg\_comments,   
   avg\_engagement\_rate,   
   RANK() OVER (ORDER BY avg\_engagement\_rate DESC, total\_followers DESC) AS influencer\_rank  
   FROM   
   UserEngagement  
   ORDER BY   
   avg\_engagement\_rate DESC, total\_followers DESC;  
     
   **OUTPUT:  
   A screenshot of a computer

   AI-generated content may be incorrect.**

**Insight:** These are the below users who has the more followers and high engagement rates Ideal for influencing and marketing campaigns.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Username | followers | posts | avg\_likes | avg\_comments | total | rank |
| Meggie\_Doyle | 76 | 1 | 41 | 34 | 152 | 1 |
| Cesar93 | 77 | 10 | 33.8 | 30.8 | 151.6 | 2 |
| Eveline95 | 77 | 12 | 35 | 27.4167 | 151.4167 | 3 |
| Aurelie71 | 77 | 8 | 35 | 30.25 | 150.25 | 4 |
| Jaylan.Lakin | 76 | 1 | 38 | 35 | 150 | 5 |

**Recommendation:**

* + We can pay the influencers to promote the brand.
  + We can send freebies or free coupon codes of the brand to influencers they can redeem it to buy.
  + Interacting to their posts and reels continuously.
  + We can DM to them directly and ask them to collaborate and work with them for promoting the brand.
  + Explaining how the partnership benefits them.
  + We can also measure the ad campaigns impact on engagement, clicks and conversions.

1. **Based on user behavior and engagement data, how would you segment the user base for targeted marketing campaigns or personalized recommendations?  
     
   SQL QUERY:**with UserActivity as (  
   select  
   u.id as user\_id,  
   u.username,  
   count(distinct f.follower\_id) as total\_followers,   
   count(distinct p.id) as total\_posts,   
   coalesce(avg(like\_count), 0) as avg\_likes,   
   coalesce(avg(comment\_count), 0) as avg\_comments,  
   (coalesce(avg(like\_count), 0) + coalesce(avg(comment\_count), 0) ) as avg\_engagement\_rate  
   from users u  
   left join follows f ON u.id = f.followee\_id -- Count followers  
   left join photos p ON u.id = p.user\_id -- Count posts  
   left join (  
   select photo\_id, count(user\_id) as like\_count from likes group by photo\_id  
   ) l ON p.id = l.photo\_id  
   left join (  
   select photo\_id, count(id) as comment\_count from comments group by photo\_id  
   ) c on p.id = c.photo\_id  
   group by u.id, u.username  
   )  
   select   
   user\_id,   
   username,   
   total\_followers,   
   total\_posts,   
   avg\_likes,   
   avg\_comments,   
   avg\_engagement\_rate,  
   case   
   WHEN total\_followers >= 70 AND avg\_engagement\_rate >= 65 THEN 'Influencer'  
   WHEN total\_followers >= 60 AND avg\_engagement\_rate >= 55 THEN 'Highly Engaged Users'  
    ELSE 'Inactive User'  
   END AS user\_segment  
   FROM   
   UserActivity  
   ORDER BY avg\_engagement\_rate DESC;  
     
   **OUTPUT:  
   A screenshot of a computer

   AI-generated content may be incorrect.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | Username | total\_followers | total\_posts | avg\_likes | avg\_comments | avg\_engagement\_rate | user\_segment |
| 55 | Meggie\_Doyle | 76 | 1 | 41 | 34 | 75 | Influencer |
| 73 | Jaylan.Lakin | 76 | 1 | 38 | 35 | 73 | Influencer |
| 48 | Granville\_Kutch | 76 | 1 | 37 | 34 | 71 | Influencer |
| 22 | Kenneth64 | 76 | 1 | 39 | 31 | 70 | Influencer |

**Personalized content strategies:**

**Influencers:** Prioritize for brand partnerships & collaborations.

**Highly Engaged Users:** Suggest new features, trending hashtags and community events.

**Inactive Users:** Send personalized re-engagement messages.

**Optimized content Strategies:**

**Retargeting Inactive Users:** Personalized discounts or promotions.

**Interest-Based Ads:** Match ads with their past engagement behaviour.

**Gamification and Rewarding System:**

* + Leaderboard Ranking for highly active users.
  + Loyalty points for consistent reactions.
  + Special badges for content creations.

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

Will consider the below metrics to measure their effectiveness:

* + **Click-Through Rate (CTR):** Calculate the ratio of clicks to impressions to gauge how compelling your ad is.
  + **Conversion Rate:** Measure the percentage of clicks that result in a conversion to assess the ad’s ability to drive action.
  + **Cost Per Click (CPC) & Cost Per Acquisition (CPA):** Monitor spending efficiency by calculating the cost for each click or conversion.
  + **Return on Ad Spend (ROAS):** Evaluate revenue generated per dollar spent to determine overall profitability.
  + Analyse time-of-day and day-of-week performance to determine when your target audience is most active.
  + Adjust the timing and frequency of your ads to maximize visibility without causing ad fatigue.
  + Track the customer journey from ad impression through click to conversion.
  + **Feedback:** Regularly review performance data and gather qualitative feedback to understand why certain ads work better.
  + Use these insights to continuously iterate on creative elements, targeting strategies, and budget allocation.

1. **How can you use user activity data to identify potential brand ambassadors or advocates who could help promote Instagram's initiatives or events?  
     
   SQL QUERY:**with UserActivity as (  
   select  
   u.id as user\_id,  
   u.username,  
   count(distinct f.follower\_id) as total\_followers,   
   count(distinct p.id) as total\_posts,   
   coalesce(avg(like\_count), 0) as avg\_likes,   
   coalesce(avg(comment\_count), 0) as avg\_comments,  
   (coalesce(avg(like\_count), 0) + coalesce(avg(comment\_count), 0) ) as avg\_engagement\_rate  
   from users u  
   left join follows f ON u.id = f.followee\_id -- Count followers  
   left join photos p ON u.id = p.user\_id -- Count posts  
   left join (  
   select photo\_id, count(user\_id) as like\_count from likes group by photo\_id  
   ) l ON p.id = l.photo\_id  
   left join (  
   select photo\_id, count(id) as comment\_count from comments group by photo\_id  
   ) c on p.id = c.photo\_id  
   group by u.id, u.username  
   )  
     
   select   
   user\_id,  
   username,  
   total\_followers,   
   total\_posts,   
   avg\_likes,   
   avg\_comments,   
   avg\_engagement\_rate,   
   case   
   WHEN total\_followers >= 70 AND avg\_engagement\_rate >= 65 THEN 'High-Engagement Influencer'  
   WHEN total\_followers >= 60 AND avg\_engagement\_rate >= 55 THEN 'Regular User'  
   ELSE 'Inactive User'  
   END AS potential\_ambassador  
   FROM   
   UserActivity  
   ORDER BY avg\_engagement\_rate DESC;  
     
   **OUTPUT:  
   A screenshot of a graph

   AI-generated content may be incorrect.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | Username | total\_followers | total\_posts | avg\_likes | avg\_comments | avg\_engagement\_rate | user\_segment |
| 55 | Meggie\_Doyle | 76 | 1 | 41 | 34 | 75 | High-Engagement Influencer |
| 73 | Jaylan.Lakin | 76 | 1 | 38 | 35 | 73 | High-Engagement Influencer |
| 48 | Granville\_Kutch | 76 | 1 | 37 | 34 | 71 | High-Engagement Influencer |
| 22 | Kenneth64 | 76 | 1 | 39 | 31 | 70 | High-Engagement Influencer |

**To Promote Instagram Initiatives or events:**

* + Craft personalized messages that acknowledge their influence and invite them to collaborate on campaigns or events.
  + Offer incentives such as exclusive access to new features, special discounts, or monetary rewards. For example, you might propose a partnership where they receive early access to new app features in exchange for promoting them.
  + Analyse the topics and content types that drive high engagement for these users. Collaborate on campaigns that align with these interests to ensure authenticity and higher conversion rates.
  + Create a structured program (e.g., ambassador tiers) that rewards influencers based on their ongoing engagement metrics, such as continued high engagement rates or an increase in followers over time.
  + Engage these ambassadors in providing feedback on new initiatives, ensuring that your campaigns are aligned with what resonates with your most active and influential users.

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

* Begin by consulting business leaders or marketing teams to clarify what they aim to achieve with the data (e.g., increased engagement, higher conversion rates, or improved user retention).
* It will ensure the analysis directly addresses the core objectives and provides actionable insights.
* Decide on the most relevant metrics for measuring success, such as engagement rates, conversion rates, or user retention.
* It will establish clear targets to track progress, allowing us to measure the effectiveness of the campaigns or strategies.
* Review all available tables—such as users, photo tags, photos, likes, comments, follows, and tags—to understand the scope, completeness, and quality of the data.
* I will identify potential data gaps or inconsistencies early, preventing inaccurate conclusions later in the analysis.
* Analyse which content types (photos, videos, reels) or hashtags generate the highest engagement (likes, comments, shares).
* It will help prioritize content creation and promotion efforts to maximize user interest and interactions.
* Look for natural patterns or segments in user behaviour. For instance, some users may consistently attract more likes and comments, indicating potential influence.
* This will enable targeted strategies, such as partnering with high-engagement users or tailoring content to specific audience segments.
* Based on user activity data, categorize users by how frequently they post or engage. This makes it easy to identify both power users and those who need re-engagement tactics.
* This will facilitate personalized outreach—active users might receive special perks, while inactive users might get targeted campaigns to encourage them to return.
* Derive straightforward metrics—like total posts, likes, comments, and follower counts—and then compute ratios (e.g., engagement per post or follower-to-engagement ratio).
* It will simplify comparisons across users or content types, highlighting who or what drives the most interaction.
* Present initial findings to stakeholders and gather their input to refine the analysis or explore new angles.
* This will ensure your work remains relevant and actionable, leading to more effective data-driven decisions.
* Use our analysis to recommend tactics, such as personalized content recommendations or targeted ad campaigns. Validate these strategies with the client or internal teams.
* It bridges the gap between data insights and practical marketing actions, making it easier to implement improvements.
* Summarize the entire process, key assumptions, and insights in a clear report or dashboard.
* This will ensure stakeholders can easily review, share, and act on your analysis, creating a reference point for ongoing improvements and future projects.

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?**

**Approach:**

This below Query will be used to change the Engagement\_Type from “Like” to “Heart”

Update User\_Interactions   
Set Engagement\_Type = “Heart”  
where Engagement\_Type = “Like”

The 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.