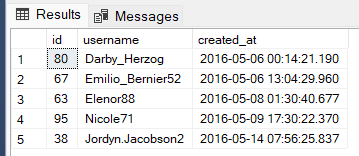
Scenario-based Challenges (Basic to intermediate)

Q1. We want to reward our users who have been around the longest.

Find the 5 oldest users.

select top 5 \* from users order by created\_at asc



Q.2 What day of the week do most users register on?

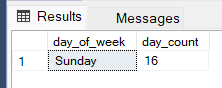
We need to figure out when to schedule an ad campgain

select top 1 DATENAME(WEEKDAY, created\_at) as day\_of\_week, count(DATENAME(WEEKDAY, created\_at)) as day\_count

from users

group by DATENAME(WEEKDAY, created\_at)

order by day\_count desc



Q3. We want to target our inactive users with an email campaign.

Find the users who have never posted a photo

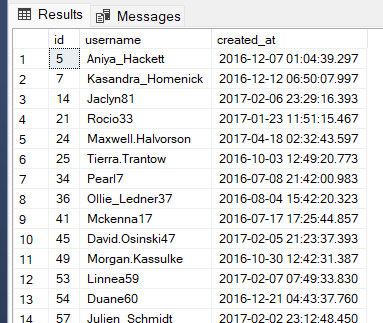
solution 1:

select \* from users where id not in (select user\_id from photos)

solution 2:

select u.\* from users u left join photos p on u.id = p.user\_id

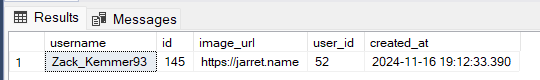
where p.id is null



Q4. We're running a new contest to see who can get the most likes on a single photo. find the user who has most liked photo?

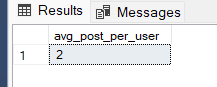
select u.username, p.\* from photos p join users u on u.id = p.user\_id

where p.id in (select top 1 photo\_id from likes group by photo\_id order by count(photo\_id) desc)



Q5. How many times does the average user post?

select count(\*) / (select count(\*) from users) as avg\_post\_per\_user from photos



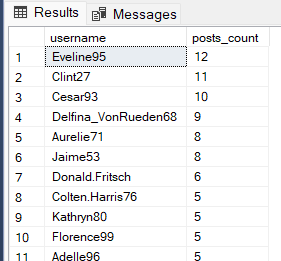
Q6. Get the list of users rankings by posting higher to lower

select u.username, count(user\_id) as posts\_count from photos p join users u

on u.id = p.user\_id

group by p.user\_id, u.username

order by posts\_count desc

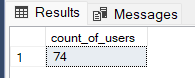


Q7. Total Posts by users (longer version of SELECT COUNT(\*) FROM photos)

total numbers of users who have posted at least one time

select count(distinct p.user\_id) as count\_of\_users

from users u join photos p on u.id = p.user\_id



Q8. A brand wants to know which hashtags to use in a post

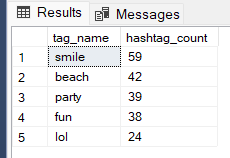
What are the top 5 most commonly used hashtags?

select top 5 t.tag\_name, COUNT(tag\_id) as hashtag\_count

from photo\_tags p join tags t on p.tag\_id = t.id

group by tag\_id, t.tag\_name

order by hashtag\_count desc



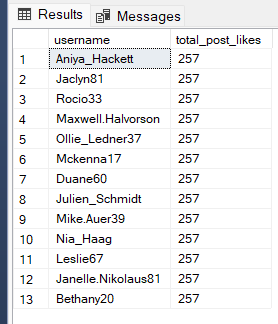
Q9. Find users who have liked every single photo on the site

select u.username, count(user\_id) as total\_post\_likes

from likes l join users u on l.user\_id = u.id

group by user\_id, u.username

having count(user\_id) = (select count(\*) from photos)



Q10. We also have a problem with celebrities

Find users who have never commented on a photo

select username from users

where id not in (select distinct c.user\_id from comments c)



Advanced Scenario questions

Q11. Are we overrun with bots and celebrity accounts?

Find the percentage of our users who have either never commented on a photo or have commented on every photo

WITH CTE as

(

select count(user\_id) as count\_users\_commented from

(select user\_id from comments

group by user\_id having count(user\_id) = (select count(\*) from photos)) as T1

),

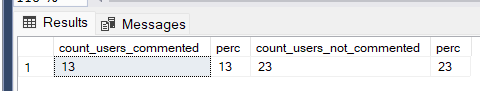
CTE2 as

(select count(id) as count\_users\_not\_commented from users where id not in (select distinct user\_id from comments))

select CTE.count\_users\_commented, round(CTE.count\_users\_commented \* 100 / (select count(\*) from users), 2) as perc,

CTE2.count\_users\_not\_commented, round(CTE2.count\_users\_not\_commented \* 100 / (select count(\*) from users), 2) as perc

from CTE join CTE2 on 1=1



Q12. Are we overrun with bots and celebrity accounts?

Find the percentage of our users who have either never commented on a photo or have liked on every photo

WITH CTE as

(

select count(\*) as tot\_users\_count\_not\_commented from users where id not in (select user\_id from comments)

),

CTE2 as

(

select count(\*) as tot\_users\_like\_all\_photo from

(select user\_id, count(user\_id) as tot\_users\_like\_all\_photo from likes

group by user\_id having count(user\_id) = (select count(\*) from photos)) as T1

)

select tot\_users\_count\_not\_commented, round(tot\_users\_count\_not\_commented \* 100/ (select count(\*) from users), 2) as perc,

tot\_users\_like\_all\_photo, round(tot\_users\_like\_all\_photo \* 100/ (select count(\*) from users), 2) as perc

from CTE join CTE2 on 1=1

