

Question(1) Create an ER diagram or draw a schema for the given database?

Entity - Relationship

Question (2) We want to reward the user who has been around the longest, Find the 5 oldest users?

ANSWER -

```
select username, created_at from users group by username  
order by created_at limit 5;
```

```
select username as 5_oldest_users, created_at from  
(  
select * from users order by created_at  
) users  
order by created_at limit 5;
```

Question (3) To understand when to run the ad campaign, figure out the day of the week most users register on ?

ANSWER -

```
select dayname(created_at) as weekday, count(username) as no_of_users_register from users group by  
weekday  
order by no_of_users_register desc;
```

Question(4) To target inactive users in an email ad campaign, find the users who have never posted a photo?

ANSWER -

```
select username from users where id not in  
(select user_id from photos);
```

```
select username, p.id from users as u  
left join photos as p  
on u.id = p.user_id where p.id is null;
```

Question(5) Suppose you are running a contest to find out who got the most likes on a photo. Find out who won?

ANSWER -

```
select users.username, photos.id as photo_id, count(*) as total_number_of_likes from likes  
inner join photos on photos.id = likes.photo_id inner join users on users.id = photos.user_id group by  
photos.id order by  
total_number_of_likes desc;
```

Question(6) The investors want to know how many times does the average user post?

ANSWER -

Used round -

```
select round ((select count(*) from photos)/(select count(*) from users)) as average_users_post;
```

Without used round -

```
select ((select count(*) from photos)/(select count(*) from users)) as average_users_post;
```

Question (7) A brand wants to know which hashtag to use on a post, and find the top 5 most used hashtags?

ANSWER -

```
select tag_id, tag_name, count(tag_id) as top_5_hashtags_used_mostly from photo_tags as p inner join
tags as t
on p.tag_id = t.id
group by tag_id order by top_5_hashtags_used_mostly desc limit 5;
```

Question(8) To find out if there are bots, find users who have liked every single photo on the site?

ANSWER -

```
select u.id , u.username, count(l.user_id) as total_likes from users as u inner join likes
as l on u.id = l.user_id
group by u.id having total_likes = (select count(*) from photos);
```

Question(9) To know who the celebrities are, find users who have never commented on a photos?

ANSWER -

```
select username, comment_text from users left join comments on users.id = comments.user_id
group by users.id
having comment_text is null;
```

Question(10) now its time to find both of them together, find the users who have never commented on any photo or have commented on every photos ?

ANSWER -

```
select username, comment_text, photo_id,
(case
when comment_text is null then ' nevercommented'
when comment_text is not null then ' commented'
else 'missing value' end) as ' COMMENTED AND NEVER COMMENTED'
from users
```

```
left join comments on users.id = comments.user_id  
group by users.username;
```