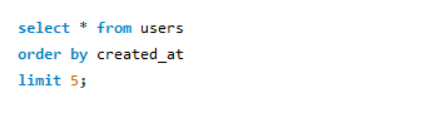
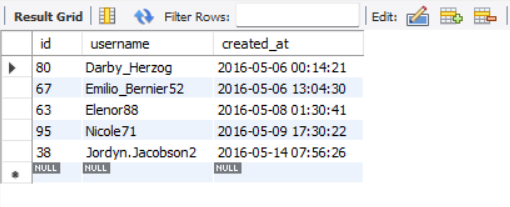
**SQL Tasks**

A) Marketing Analysis:

1. Loyal User Reward: The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

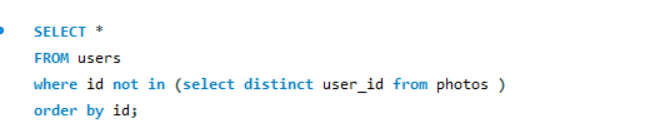
Your Task: Identify the five oldest users on Instagram from the provided database.

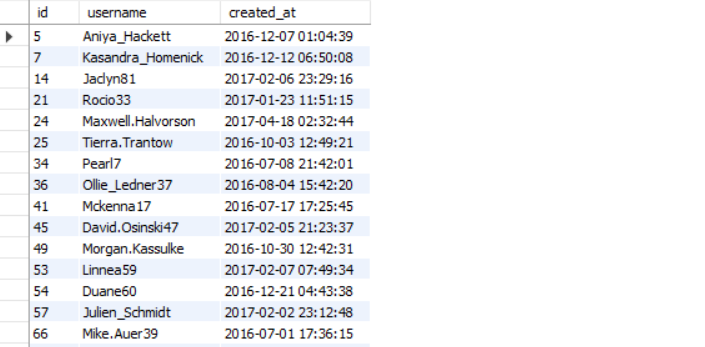


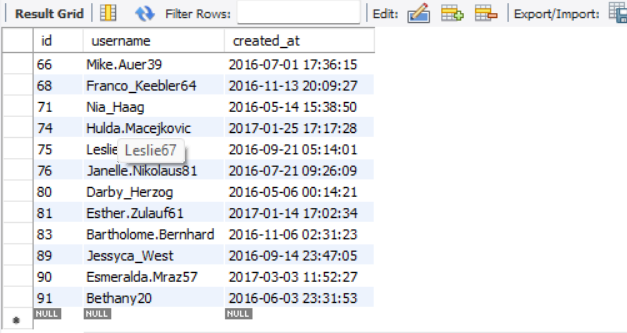


1. Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.

Your Task: Identify users who have never posted a single photo on Instagram.

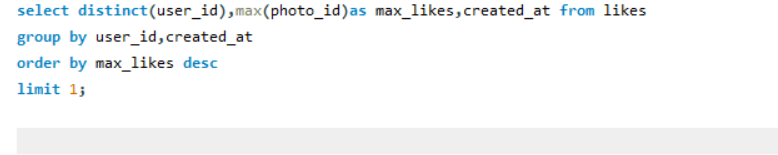


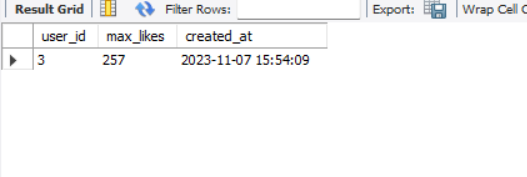




1. Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.

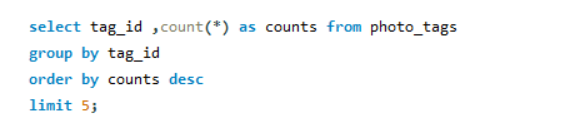
Your Task: Determine the winner of the contest and provide their details to the team.

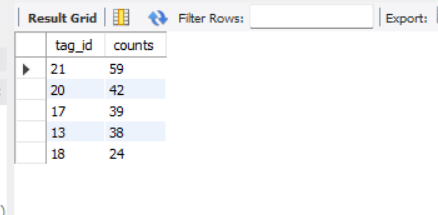




1. Hashtag Research: A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

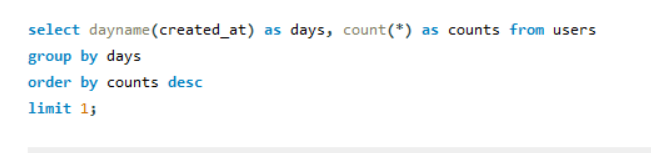
Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

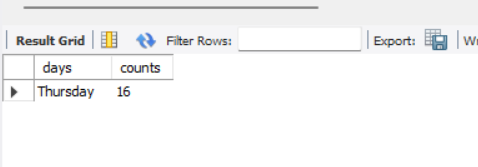




1. Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

Your Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

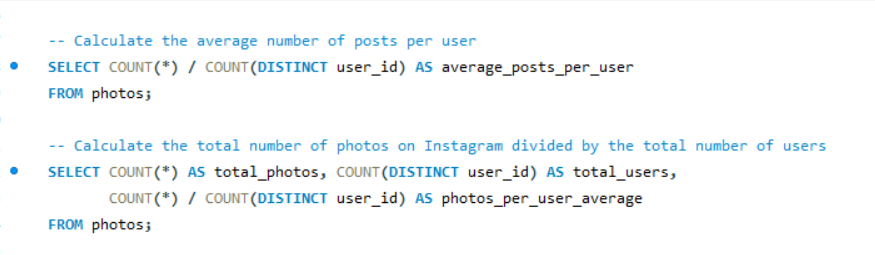


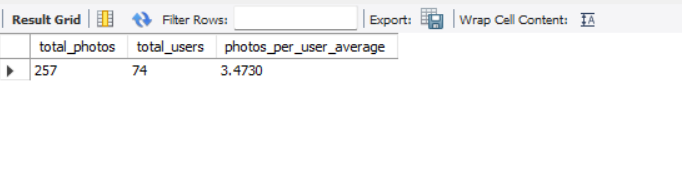


B) Investor Metrics:

1. User Engagement: Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

Your Task: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

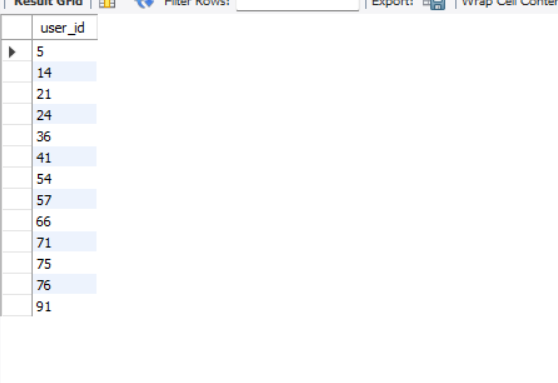




1. Bots & Fake Accounts: Investors want to know if the platform is crowded with fake and dummy accounts.

Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.





SQL Queries

use ig\_clone;

select \* from users

order by created\_at

limit 5;

SELECT \*

FROM users

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

order by id;

select user\_id,max(photo\_id)as max\_likes from likes

group by user\_id

order by max\_likes desc

limit 0;

select tag\_id ,count(\*) as counts from photo\_tags

group by tag\_id

order by counts desc

limit 5;

select dayname(created\_at) as days,count(\*) as counts from users

group by days

order by counts desc

limit 1;

-- Calculate the average number of posts per user

SELECT COUNT(\*) / COUNT(DISTINCT user\_id) AS average\_posts\_per\_user

FROM photos;

-- Calculate the total number of photos on Instagram divided by the total number of users

SELECT COUNT(\*) AS total\_photos, COUNT(DISTINCT user\_id) AS total\_users,

COUNT(\*) / COUNT(DISTINCT user\_id) AS photos\_per\_user\_average

FROM photos;

SELECT user\_id

FROM (

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

FROM likes l

JOIN photos p ON l.photo\_id = p.id

GROUP BY l.user\_id

) AS user\_likes

WHERE total\_likes = (SELECT COUNT(\*) FROM photos);