

Introduction

This SQL project involved analyzing an Instagram database to extract valuable insights related to user activity, engagement, and platform trends. The queries performed provided insights into user behavior, photo engagement, and hashtag trends. By leveraging SQL queries, different aspects of the platform were explored to inform business decisions aimed at increasing user engagement and optimizing platform performance.

Objective

The primary objectives of this project were:

1. To identify the oldest users who joined the platform.
2. To find inactive users who have never posted a photo.
3. To determine the user who won a contest based on the most-liked photo.
4. To analyze the most commonly used hashtags on the platform.
5. To determine the best day to launch ads based on user sign-ups.
6. To calculate the average number of posts per user.
7. To identify potential bot users who have liked every single photo.

Insights and Analysis

1. Identifying Oldest Users

- The top five oldest users were retrieved by sorting the user data in ascending order based on the `created_at` field.
- This information can be useful for recognizing and rewarding loyal users or studying early adoption patterns.

```
-- oldest users
• select *
  from users
 order by created_at
 limit 5;
```

2. Identifying Inactive Users

- Users who have never posted a single photo were identified by performing a **LEFT JOIN** between the **users** and **photos** tables and filtering for users without associated photos.
- These inactive users could be targeted with promotional emails encouraging them to start posting and engaging with the platform.

```
-- never posted a single photo on Instagram--
select u.id, u.username
  from users as u
 left join photos as p ON p.user_id = u.id
 where p.user_id is NULL;
```

3. Identifying the Contest Winner

- A contest was conducted where the user with the most likes on a single photo was declared the winner.
- Using a Common Table Expression (CTE), the total likes per photo were counted, and the photo with the highest number of likes was selected.
- By joining the **users** table, the winner's username and their winning photo details were identified.
- This insight can be leveraged to analyze engagement trends and replicate successful content strategies in future contests.

```

with cte as (
  SELECT photo_id, COUNT(user_id) as total_likes
FROM likes
GROUP BY photo_id
ORDER BY total_likes DESC LIMIT 1)
select u.username, u.id, p.id as photo_id, cte.total_likes from cte
join photos as p on cte.photo_id = p.id
join users as u on p.user_id = u.id;

```

4. Analyzing the Most Commonly Used Hashtags

- To determine the top five most frequently used hashtags, the `photo_tags` table was analyzed by grouping and counting occurrences of each tag.
- Understanding hashtag trends helps in identifying popular content themes and optimizing hashtag recommendations for users.

```

-- top five most commonly used hashtags the platform--
select * from tags;
with cte as (select tag_id from photo_tags
group by tag_id
order by count(tag_id) desc limit 5)
select t.tag_name from cte
join tags as t on cte.tag_id = t.id;

```

5. Determining the Best Day to Launch Ads

- The distribution of user sign-ups was analyzed by extracting the weekday from the `created_at` field and counting the number of new users per day.
- The day with the highest number of new user registrations indicates an optimal time for launching advertisements to maximize visibility.

```
select dayname(created_at) as day_week, count(*) as total_users
from users
group by dayname(created_at)
order by total_users desc;
```

6. Calculating the Average Number of Posts Per User

- The total number of photos on the platform was computed and divided by the total number of users to determine the average number of posts per user.
- Understanding this metric helps in setting engagement benchmarks and monitoring user activity levels.

```
-- 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--
select avg(total_photos) from
(select u.id, coalesce(count(p.id),0) as total_photos
from users as u left join photos as p ON p.user_id = u.id
group by u.id) as a;

select avg(total_photos),max(a) from (select distinct u.id, coalesce(count(p.id) over(partition by u.id),0) as total_photos,
round(count(p.id) over() * 1.00/ count(u.id) over(),1) as a
from users as u left join photos as p ON p.user_id = u.id) as b;
```

7. Identifying Potential Bot Users

- Users who have liked every single photo on the platform were identified by joining the `users` and `likes` tables and counting the number of likes per user.
- Comparing these counts with the total number of photos helped detect users with unusually high engagement patterns, which could indicate bot activity.

```
select username, count(*) as number_likes
from users as u join likes as l ON u.id = l.user_id
group by l.user_id
having number_likes = (select count(*) from photos)
```

Conclusion

This SQL-based analysis provided key insights that can enhance user engagement on the Instagram clone platform. Identifying inactive users allows for the implementation of strategies to encourage activity. Additionally, analyzing engagement patterns through contests, hashtags, and optimal ad launch days can help drive more interactions. Detecting potential bot users ensures platform authenticity and user trust. Future enhancements could include analyzing follower growth trends, identifying influencers, and monitoring the effectiveness of various engagement campaigns.