Instagram User Analytics

Project Description

This project uses SQL to analyze user interactions and engagement with the Instagram app. The primary goal is to extract meaningful insights that can help in making business decisions related to marketing strategies, user engagement, and investor metrics.

Tech-Stack Used

- **MySQL Workbench**
- **SQL**

Approach

I have used SQL to extract the data from the database by implementing queries using various functions, joins, nested queries to present the answers for the questions and derive insights from the results.

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.

   To identify the oldest users on Instagram we need to query on users table and order the created_at column in ascending and limit it to 5 to have 5 rows only.

   Query:

   ```
   SELECT
       *
   FROM
       users
   ORDER BY created_at
   LIMIT 5;
   ```

2. **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 Identify users who have never posted a single photo on Instagram.

Here, I implemented left join on users and photos table to get the id, username and match the id in the user table with the user_id in the photo table. Next, I implemented the where condition to attain the id that were null i.e not present in the table.

Query:

SELECT

    users.id, users.username, photos.id

FROM

    users

      LEFT JOIN

    photos ON users.id = photos.user_id

WHERE

    photos.id IS NULL;

id	username	photo
5	Aniya_Hackett	NULL
7	Kasandra_Homenick	NULL
14	Jaclyn81	NULL
21	Rocio33	NULL
24	Maxwell.Halvorson	NULL
25	Tierra.Trantow	NULL
34	Pearl7	NULL
36	Ollie_Ledner37	NULL
41	Mckenna17	NULL
45	David.Osinski47	NULL
49	Morgan.Kassulke	NULL
53	Linnea59	NULL
54	Duane60	NULL
57	Julien_Schmidt	NULL

26 rows returned.

3. 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.

I followed the nested join procedure to get the details of the winner of the contest. I joined user table with the photo table to get the user_id and photo_id.

Then joined it with likes table to get the likes on the particular photo_id and then grouped it by photo id to gat the number of likes and then ordered it descending to get the highest liked photo.

Query:

```
SELECT

        u.id AS user_id,

        u.username,

        p.id AS photo_id,

        p.image_url,

        COUNT(l.photo_id) AS num_likes

FROM

        users u

JOIN

        photos p ON u.id = p.user_id

JOIN

        likes l ON p.id = l.photo_id

GROUP BY

        p.id

ORDER BY

        num_likes DESC;

        LIMIT 1;
```

user_id	username	photo_id	image_url	num_likes
52	Zack_Kemmer93	145	https://jarret.name	48

4. **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.

To get the top five commonly used tag I implemented join on tag table and photo_tags table.

Each tag has a unique id in tag table and different photos have these tag associated in photo_tag table. I grouped the tag id's in the photo_tag table to get the count of frequently used tag.

Query:

```
SELECT
    t.id, t.tag_name, COUNT(pt.tag_id) AS frequency
FROM
    tags t
        JOIN
    photo_tags pt ON t.id = pt.tag_id
GROUP BY t.id
ORDER BY frequency DESC
LIMIT 5;
```

id	tag_name	frequency
21	smile	59
20	beach	42
17	party	39
13	fun	38
18	concert	24

5. **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.

    To get the day at which most users register on the platform, I used Dayof week method to extract the day and then grouped them based on weekdays to count the users using count().

    Query:

    ```
    SELECT
        DAYOFWEEK(created_at) AS weekdays,
        COUNT(id) AS registered_users
    FROM
        users
    GROUP BY weekdays
    ORDER BY registered_users DESC;
    ```

Result Grid		Filter Rows:
weekdays	registered_users
5	16
1	16
6	15
3	14
2	14
4	13
7	12

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.

   I solved the problem by breaking down it into 2 queries

   1. Calculate the average number of posts per user on Instagram.
   2. Also, provide the total number of photos on Instagram divided by the total number of users.

   Then I implemented the nested query to obtain the result.

   I implemented the join operation on inner query to get the users and photos posted then implemented the sum() to get the total posts and count() using distinct to get the total users then divided the expression to get the avg post.

   Query:
   ```
   SELECT
       sum(posted)as total_posts,
       count(distinct total_users) as total_users
       SUM(posted) / COUNT(DISTINCT total_users) AS avg_post
   FROM
       (SELECT
           users.id AS total_users, COUNT(photos.id) AS posted
       FROM
           users
       LEFT JOIN photos ON photos.user_id = users.id
       GROUP BY users.id) AS avg;
   ```

Result Grid	Filter Rows:		
	total_posts	total_users	avg_post
▶	257	100	2.5700

2. **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.

To get the potential bots based on the likes, I applied the inner join on likes and user table to get the users who have liked each and every photo using the count() to get the count of likes and having method to get the users who have liked all photos.

Query:
SELECT

    user_id, username, count(*) as total_likes

FROM

    likes

       INNER JOIN

    users ON likes.user_id = users.id

GROUP BY user_id

HAVING total_likes = (SELECT

       COUNT(*)

   FROM

      photos);

user_id	username	total_likes
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

Insights

1. Identified the top 5 loyal users who have been using the platform for longest time.
   All the top 5 users registered on may 2016.

2. There were 26 users who have been inactive. Promotional emails must be send to those 26 members to encourage them to post.
3. User_id 52 Zack_kemmer93 won the contest have received 48 likes for photo_id 145.
4. The top 5 hashtags to be used to reach the most people are smile, beach, party, fun, concert.
5. Analyzed that Sunday and Thursdays are the best days to publish ads as most user's register on the platform on that particular days.
6. The average number of post per user is 3. The investor can get to know the inactive user by calculating post per user and user's who has posted less than 3 photos can be taken as inactive.
7. The total users who have liked photos are 13. Investor can note that out of 100 there are 13 users who are potential bots.

Result:

The project provided valuable insights into user behaviour and engagement on Instagram.

By identifying loyal users, inactive users, popular hashtags, optimal ad campaign times, and potential bot accounts, we can help improve user experience and engagement on the platform.

The analysis highlights key areas for marketing and development teams to focus on, ensuring targeted strategies for retaining the inactive user and encouraging new users to join.