# Analyzing Instagram User Data Using SQL

## **Introduction:**

In this project , I assumed the role of a data analyst working with Instagram's project team. My task was to analyze user interactions and engagement on the platform , providing valuable insights for marketing strategies , product development , and investor reports. This involved writing and executing SQL queries to extract meaningful data from s simulated Instagram database.

# **Project objectives**

- 1. Marketing Analysis:
  - Reward the most loyal users.
  - Engage inactive users.
  - Determine the contest winner.
  - Research the most popular hash tags.
  - Find the best day to launch ad campaigns.

## 2. Investor Metrics:

- Asses user management.
- Identify potential bots or fake accounts.

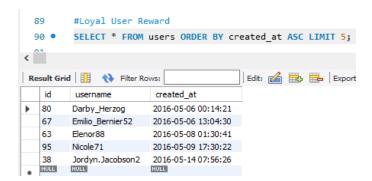
#### **Database Schema Overview**

This project utilized a database with the following tables:

- users: Stores user information.
- photos : Contains details about user photos.
- comments: tracks comments on photos.
- likes: Records user likes on photos.
- follows: Captures follower-followers relationships.
- tags: Maintains unique tags used on the platform.
- photo\_tags: Connects photos with their respective tags.

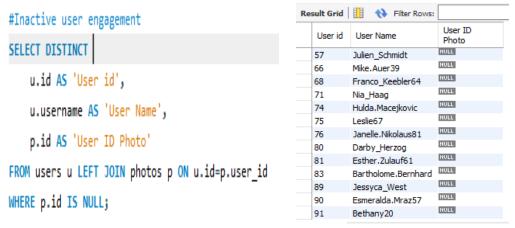
# **Key SQL Queries and Snapshots:**

## 1. Loyal User Reward:



Identified the first five users who joined the Instagram, highlighting the most loyal users based on their registration dates.

# 2. Inactive User Engagement:



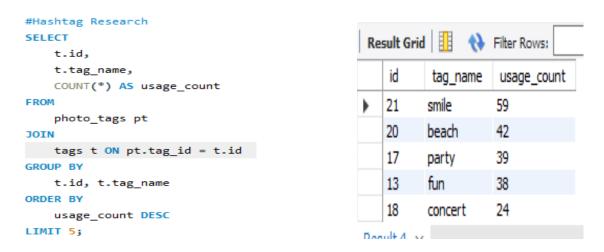
Identified users who have never posted a photo , providing a target list for re – engagement campaigns.

#### 3. Contest Winner Declaration:

```
#Contest Winner Declaration
                                                                      114 LEFT JOIN
SELECT
                                                                              likes 1 ON p.id = 1.photo_id
    u.id AS user_id,
                                                                      116 GROUP BY
    u.username,
                                                                      117
                                                                              p.id, u.id, u.username, u.created_at, p.image_url, p.created_dat
                                                                      118 ORDER BY
    u.created_at AS user_created_at,
    p.id AS photo_id,
                                                                     119
                                                                             count_userID DESC
                                                                     120 LIMIT 1;
    p.image_url,
                                                                     121
    p.created_dat AS photo_created_at,
                                                                     122 #Hashtag Research
    p.id AS photo_id,
                                                                     123 • SELECT
    COUNT(1.user_id) AS count_userID
FROM
                                                                      :
    photos p
                                                                      Export: 📳 | Wrap Cell Content: 🏗 | Fetch rows:
JOIN
                                                                       user_id username user_created_at photo_id image_url photo_created_at photo_id count_userID
    users u ON p.user_id = u.id
                                                                             Zack_Kemmer93 2017-01-01 05:58:22 145 https://jarret.name 2024-08-19 17:48:11 145
```

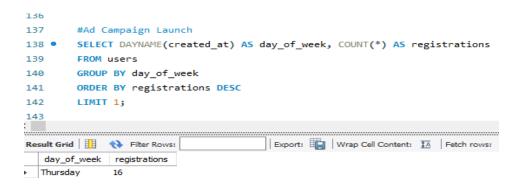
Determined the contest winner by indentifying the photo with the most likes.

## 4. Hash Tag Research:



Suggested the top five most popular hash tags to improve the brand reach on the platform .

## 5. Ad Campaign Launch:



Identified the best day to the week to launch ad campaign based on user registration trends.

## **Investor Metrics:**

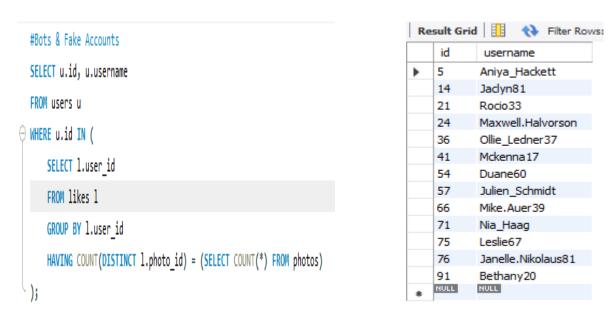
## **6. User Engagement for Investors :**

```
#User Engagement
144
        SELECT AVG(photo_count) AS avg_photos_per_user
145 •

⊖ FROM (
146
147
           SELECT user_id, COUNT(id) AS photo_count
           FROM photos
148
149
           GROUP BY user id
        ) AS user_photos;
150
Export: Wrap Ce
  avg_photos_per_user
3.4730
```

Provided insights into user engagement by calculating the average number of posts per user.

## 7. Identifying Bots & Fake Accounts:



Detected potential bot accounts by identifying users who liked every single photo on the platform.

## **CONCLUSION:**

This project allowed me to apply SQL skills to analyze real world data offering insights that could directly influence business decisions . I learned how to structure queries for various use cases , how to think critically about the data-driven strategies , and how to present findings that are both actionable and relevant to different stakeholders.

## **TOOLS USED:**

- 1. SQL
- 2. MySQL Workbench