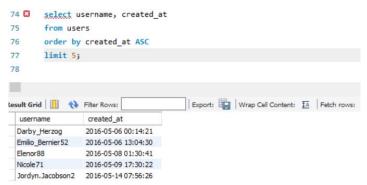
# **Instagram User Analytics**

Rukmini Annadata
 rukmini2k2@gmail.com

## **SQL Tasks:**

## A) Marketing Analysis:

#### 1.Loyal User Reward:



To identify the most loyal, i.e., the top 5 oldest users of Instagram, follow these steps:

- 1. Retrieve data from the "users" table by selecting the "username" and "created\_at" columns.
- 2. Sort the data in ascending order based on the "created" at "column using the "ORDER BY" function.
- 3. Use the "LIMIT" function to display only the top 5 oldest Instagram users in the output.

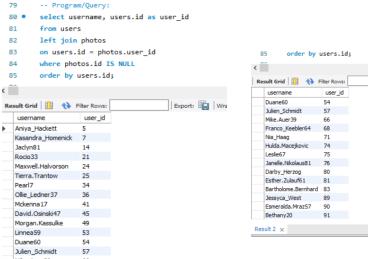
To identify the top 5 most loyal users of Instagram, we will retrieve the usernames and creation dates from the users table. The data will be sorted in ascending order based on the creation dates using the ORDER BY function. This way, we can find the users who have been using the platform for the longest time.

## 2.Inactive User Engagement

To identify the most inactive users on Instagram, those who have never posted a single photo, we will follow these steps:

- 1. Select the "username" column from the "users" table.
- 2. Perform a LEFT JOIN between the "users" table and the "photos" table, matching "users.id" with "photos.user\_id" since they share common data.
- 3. Filter the rows from the "users" table where there is no matching "photos.id" (i.e., where the "photos" table does not have any records for that user). This will give us the users who have never posted a photo on Instagram.

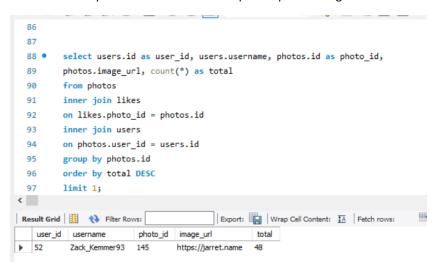
  79 --- Program/Query:



#### 3. Contest Winner Declaration:

To find the username, photo\_id, image\_url, and total\_number\_of\_likes of the most-liked photo on Instagram, we will perform the following steps:

- 1. Select the "users.username," "photos.id," "photos.image\_url," and "COUNT(\*) as total" from the respective tables.
- 2. Perform an inner join among the "photos," "likes," and "users" tables using the conditions "likes.photo\_id = photos.id" and "photos.user id = users.id."
- 3. Group the output based on "photos.id" using the GROUP BY function.
- 4. Sort the data in descending order based on the "total" using the ORDER BY function.
- 5. Retrieve only the information of the top-liked photo using the LIMIT function.



#### 4. Hashtag Research:

To determine the top 5 most commonly used hashtags on Instagram, follow these steps:

- 1. Select the "tag\_name" column from the "tags" table and use the "COUNT(\*) as total" function to count the number of times each hashtag is used individually.
- 2. Perform an inner join between the "tags" table and the "photo\_tags" table on the condition "tags.id = photo\_tags.tag\_id" as they both contain related information, i.e., "tag\_id."
- 3. Group the resulting output based on "tags.tag\_name" using the GROUP BY function.
- 4. Sort the grouped data in descending order based on the "total" (the total number of occurrences for each tag\_name) using the ORDER BY function.
- 5. Retrieve only the top 5 most used tag names by applying the LIMIT 5 function.

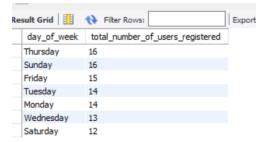


## **5.Ad Campaign Launch:**

To determine the day of the week on which most users register on Instagram, follow these steps:

- 1. Select the "dayname(created\_at) as day\_of\_week" and "count(\*) as total\_number\_of\_users\_registered" columns from the "users" table to create the desired output table.
- 2. Group the output table based on the "day\_of\_week" using the GROUP BY function, which will group the user registration data according to the day of the week.
- 3. Sort the output table in descending order based on the "total\_number\_of\_users\_registered" using the ORDER BY function. This will help us identify the day with the highest number of user registrations.

```
107
108 • select dayname(created_at) as day_of_week,
109 count(*) as total_number_of_users_registered
110 from users
111 group by day_of_week
112 order by total_number_of_users_registered DESC;
113
```



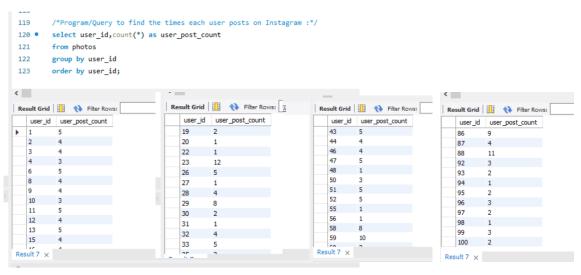
# **B) Investor Metrics:**

#### 1.User Engagement:

To calculate the average number of posts on Instagram per user, follow these steps:

- 1. Begin by counting the total number of photos (posts) in the "photos" table using the "count(\*)" function.
- 2. Similarly, count the total number of users in the "users" table using the "count(\*)" function.
- 3. Divide the total number of photos by the total number of users (count from photos / count from users). This will give us the average number of posts per user.
- 4. To find how frequently users post on Instagram, you need to determine the total occurrences of each unique "user\_id" in the "photos" table. This will help you understand the posting behavior of individual users.





## 2.Bots & Fake Accounts:

To identify potential bots and fake accounts on Instagram, follow these steps:

- 1. Select the "user\_id" column from the "photos" table to track users who have received likes on their photos.
- 2. Additionally, select the "username" column from the "users" table to display the usernames of these users.
- 3. Use the "count(\*)" function to calculate the total number of likes each user has received from the "likes" table.
- 4. Perform an inner join between the "users" and "likes" tables based on the common field "users.id" and "likes.user id" using the "on" clause.
- 5. Group the output table using the "group by" function, which will group the results based on "likes.user\_id" to consolidate the likes for each user.
- 6. Finally, search for users whose total likes (count from photos) match the total likes received (total likes per user). These users might be potential bots or accounts engaging in suspicious behavior.

```
select user_id, username, count(*) as total_likes_per_user
126 •
        from users
        inner join likes
        on users.id = likes.user_id
        group by likes.user_id
130
        having total_likes_per_user = (select count(*) from photos);
131
<
Export: Wrap Cell Content: IA
                          total likes per user
          Aniva Hackett
                         257
                   257
   14
          Jadvn81
   21
          Rocio33
                         257
        Maxwell.Halvorson 257
   24
   36
          Ollie_Ledner37
                         257
   41
         Mckenna 17
                        257
          Duane60
                         257
   57
         Julien_Schmidt 257
   66
          Mike.Auer39
                         257
   71
                        257
          Nia Haaq
   75
          Leslie67
                         257
          Janelle.Nikolaus81 257
   76
   91
          Bethany20
                         257
Result 8 ×
```

```
FOREIGN KEY (follower_id) REFERENCES users(id),
    44
                             FOREIGN KEY (followee id) REFERENCES users(id),
    45
                             PRIMARY KEY(follower_id,followee_id)
                    );
    49 • ⊖ CREATE TABLE tags(
    50
                             id INTEGER AUTO INCREMENT PRIMARY KEY,
    51
                             tag name VARCHAR(255) UNIQUE NOT NULL,
                             created_at TIMESTAMP DEFAULT NOW()
    52
    53
                     /*junction table: Photos - Tags*/
    56 • ⊖ CREATE TABLE photo_tags(
    57
                             photo id INT NOT NULL,
    58
                             tag_id INT NOT NULL,
    59
                             FOREIGN KEY(photo id) REFERENCES photos(id),
    60
                             FOREIGN KEY(tag id) REFERENCES tags(id),
    61
                             PRIMARY KEY(photo_id,tag_id)
    63 •
                    INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-02 17:11:21.4
    64 •
                    INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1), ('http://oicky.biz', 1
   65 •
                    INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10), (2, 11),
   66 •
                    INSERT INTO comments(comment text, user id, photo id) VALUES ('unde at dolorem', 2, 1), ('quae ea ducimus', 3, 1), ('alias a voluptatu
   67 •
                    INSERT INTO likes(user id, photo id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (35, 1), (36,
                     INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicious'), ('beau
    70 •
                     INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2, 2), (3, 8),
    71
<
```