INSTAGRAM USER ANALYTICS

Prachi Ranjan

Project Description

The Project is about finding out the various insights in Instagram User database. We analyze this data on the following points:

A. Marketing

1. Rewarding Most Loyal Users
2. Remind Inactive Users to Start Posting
3. Declaring Contest Winner
4. Hashtag Researching
5. Launch AD Campaign

B. Investor Metrics

1. User Engagement
2. Bots & Fake Accounts

Software used

MySQL Workbench 8.0 CE

Marketing

Rewarding Most Loyal Users: People who have been using the platform for the longest time.

Task1: Find the 5 oldest users of the Instagram from the database provided.

- We will use **select** statement to select username and created_at column from users table.
- We will use **order by** clause with **asc** in created_at column to sort the output in ascending order.
- Using **limit** function, we can display the output for top 5 oldest users of the Instagram.

QUERY:-

```
select username, created_at
from users
order by created_at asc
limit 5;
```

Task1: Find the 5 oldest users of the Instagram from the database provided.

Output/Results:-

username	created_at
Darby_Herzog	06-05-2016 00:14
Emilio_Bernier52	06-05-2016 13:04
Elenor88	08-05-2016 01:30
Nicole71	09-05-2016 17:30
Jordyn.Jacobson2	14-05-2016 07:56

Marketing

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

Task2: Identify users who have never posted a single photo on Instagram.

- We will use **select** statement to select username and user id column from users table then assign **alias** for users table as **u** and photos table as **p**
- we will **left join** photos table on users table to retrieve photo id column from photos table **on u.id= p.user_id** because both columns have common content.
- Using **where** clause we will filter rows from users table where **p.id is null**.
- We will use **order by** clause in id column in users table to sort the output.

QUERY:-

```
select u.username, u.id
from users u
left join photos p
on u.id= p.user_id
where p.id is null
order by u.id;
```

Task2: Identify users who have never posted a single photo on Instagram.

Output/Results:-

There are 26 users who are inactive on Instagram.

username	id
Aniya_Hackett	5
Kasandra_Homenick	7
Jaclyn81	14
Rocio33	21
Maxwell.Halvorson	24
Tierra.Trantow	25
Pearl7	34
Ollie_Ledner37	36
Mckenna17	41
David.Osinski47	45
Morgan.Kassulke	49
Linnea59	53
Duane60	54
Julien_Schmidt	57
Mike.Auer39	66
Franco_Keebler64	68
Nia_Haag	71
Hulda.Macejkovic	74
Leslie67	75
Janelle.Nikolaus81	76
Darby_Herzog	80
Esther.Zulauf61	81
Bartholome.Bernhard	83
Jessyca_West	89
Esmeralda.Mraz57	90
Bethany20	91

Marketing

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

Task3: Determine the winner of the contest and provide their details to the team.

- We will use **select** statement to select id, username column from users table, photo_id column from Likes table, image_url column from photos table.
- Then we will use **count** function in Likes table to count the likes and assign alias as no_of_likes.
- we will do **inner join** photos table on Likes table **on l.photo_id = p.id ,** Users table on photos table **on p.user_id = u.id** and assign alias for likes table as **l ,** photos table as **p ,** users table as **u.**
- We will use **group by** clause in photo_id column from likes table to get number of likes for each photo_id.
- We will use **order by** clause with **desc** in no_of_likes column to sort the output in descending order.
- Using **limit** function, we can display the output of user with the most likes on single photo on Instagram.

Task3: Determine the winner of the contest and provide their details to the team.

QUERY:-

```
select u.id, u.username, l.photo_id, p.image_url, count(*) as no_of_likes
from likes l
inner join photos p
on l.photo_id = p.id
inner join users u
on p.user_id = u.id
group by l.photo_id
order by no_of_likes
desc limit 1;
```

Task3: Determine the winner of the contest and provide their details to the team.

Output/Results:-

id	username	photo_id	image_url	no_of_likes
52	Zack_Kemmer93	145	https://jarret.name	48

So Zack_Kemmer93 with user id 52 is the winner because he has most number of likes i.e 48 on his single photo with photo_id 145.

Marketing

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

Task4: Identify and suggest the top five most commonly used hashtags on the platform.

- We will use **select** statement to select tag_name column from Tags table and use **count** function to count the number of times tags used then assign **alias** for Tags table as **t** and photo_tags table as **pt**
- we will **left join** tags table on photo_tags table **on pt.tag_id= t.id** because both columns have common content.
- Using **group by** clause we will get number of times tags used for each tag_id.
- We will use **order by** clause with **desc** in no_of_times_used column to sort the output in descending order.
- Using **limit** we will get top five most commonly used hashtags.

Task4: Identify and suggest the top five most commonly used hashtags on the platform.

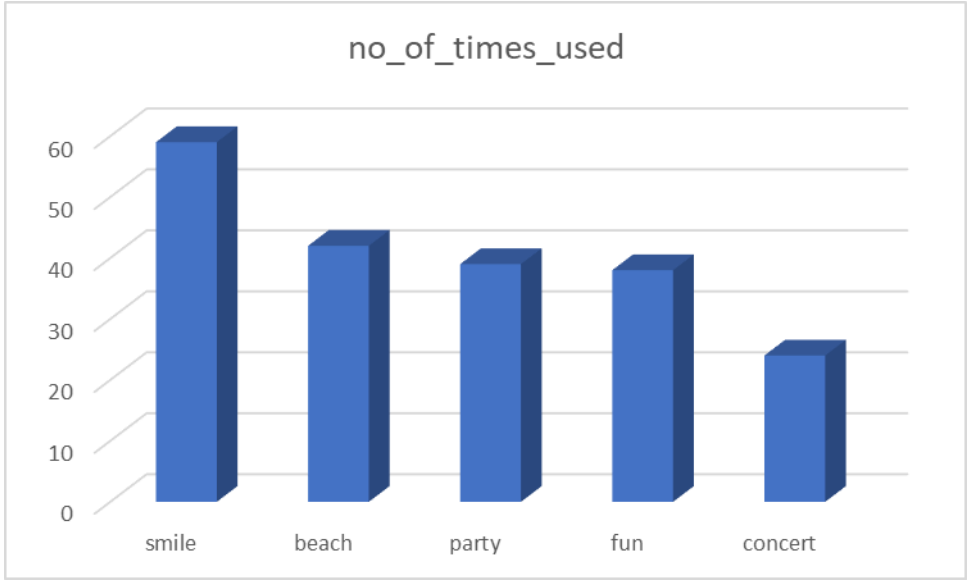
QUERY:-

```
select t.tag_name, count(*) as no_of_times_used
from photo_tags pt
left join tags t
on pt.tag_id= t.id
group by pt.tag_id
order by no_of_times_used desc
limit 5;
```

Task4: Identify and suggest the top five most commonly used hashtags on the platform.

Output/Results:-

tag_name	no_of_times_used
smile	59
beach	42
party	39
fun	38
concert	24



Marketing

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

Task5: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

- We will create columns for desired output using **select dayname(created_at) as day_of_week** and **count(*) as no_of_users_registered** from users table.
- Using **group by** clause in day_of_week column to get number of users registered in each day of the week.
- Then We will use **order by** clause with **desc** in no_of_users_registered column to sort the output in descending order.

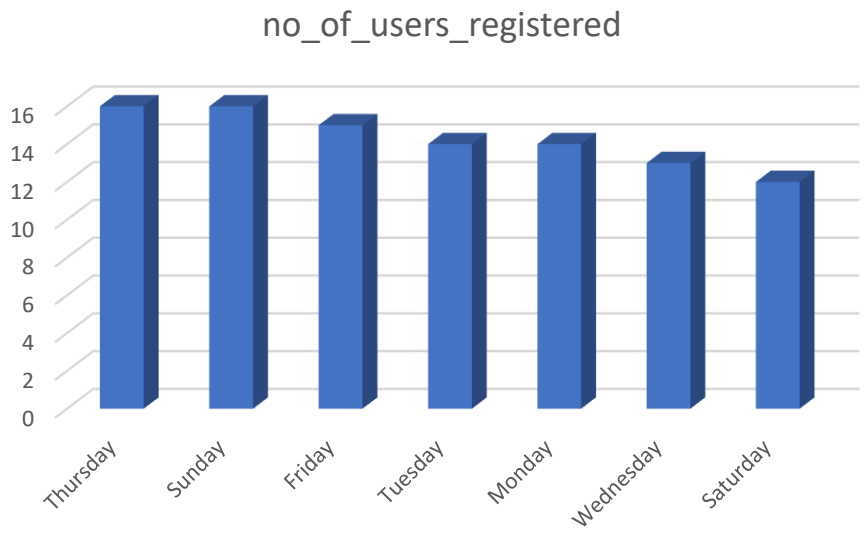
QUERY:-

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

Task5: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

Output/Results:-

day_of_week	no_of_users_registered
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12



Most of the users registered on Thursday and Sunday i.e 16. So best day of the week to launch ads are Thursday and Sunday.

Investor Metrics

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

Task1: 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.

- We will use **select** statement and **count** function to count as total and to count **distinct** user_id from photos table and divide them then we will get Avg_num_of_post_per_user.
- We will use select statement for subquery and **count** function as total from photos table and as total from users table and divide them
- Then we will get the total number of photos on Instagram divided by the total number of users.

QUERY:-

```
select count(*)/count(distinct user_id) as Avg_num_of_post_per_user

from photos;
```

Output/Results:-

Avg_num_of_post_per_user
3.473

There are total 257 photos and total 74 distinct user in photo table which gives desired output 257/74 =3.47. Average number of posts per user is 3.

Task1: 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.

QUERY:-

```
select ((select count(*) from photos)/ (select count(*) from users)) as
total_no_of_photos_divide_total_no_of_users;
```

Output/Results:-

total_no_of_photos_divide_total_no_of_users
2.57

There are total 257 photos in photos table and 100 user in users table which gives desired output is 257/100 =2.57

Investor Metrics

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

Task2: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

We will use **select** statement to select user_id from Likes table and username from users table.

- We will use **count** function in **distinct** photo_id from Likes table and assign alias for Likes table as **l** and users table **u.**
- Then we will **left join** Users table on Likes table
- Using **group by** clause we will get output for each user.
- Then we will use **having** function to search values count(*) from photos equal values to total_no_of_photos_liked.

Task2: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

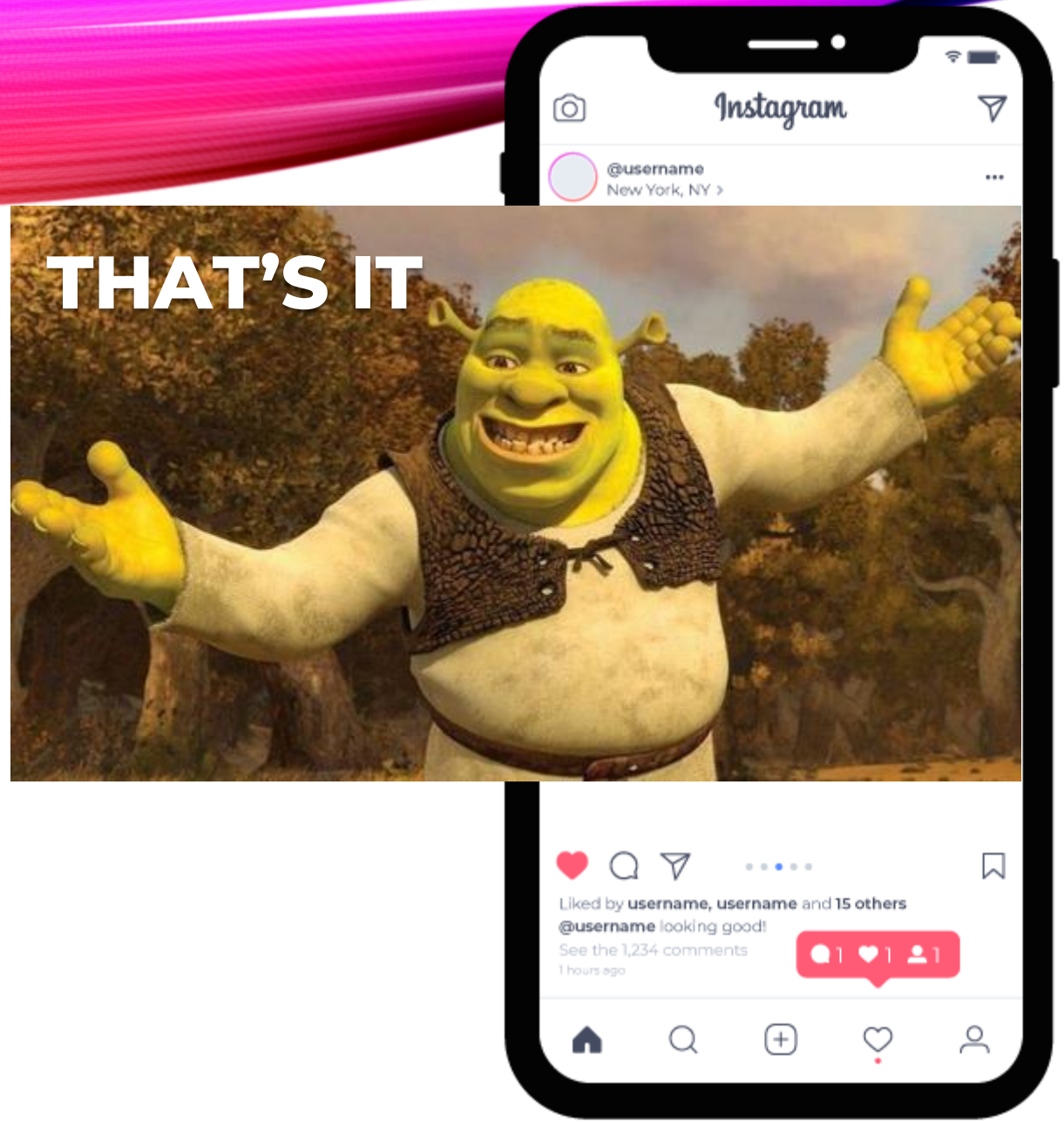
QUERY:-

```
select l.user_id, u.username, count(distinct l.photo_id) as total_no_of_photos_liked
from likes l
left join users u
on l.user_id = u.id
group by l.user_id
having total_no_of_photos_liked = (select count(*) from photos)
```

Output/Results:-

user_id	username	total_no_of_photos_liked
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

There are 30 users who have liked every single photo on the site. They can be identified as bots or fake accounts.



Thanks

Questions?