Instagram User Analytics

I am working with the product team of Instagram and the product manager has told to provide insights on the questions asked by the management team.

Analysis done with the following questions:

A) Marketing: The marketing team wants to launch some campaigns, and they need the following answers:
   a. People who have been using the platform for the longest time (Rewarding most loyal users)
   b. Users who have never posted a single photo on Instagram (Remind inactive users to start posting)
   c. Identify the winner of the contest and provide their details to the team (Declaring contest winner)
   d. Identify and suggest the top 5 most commonly used hashtags on the platform (Hashtag researching)
   e. What day of the week do most users register on? Provide insights on when to schedule an ad campaign (Launch AD campaign)

B) Investor Metrics: Investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds.
   a. Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users. (User engagement)
   b. Provide data on users (bots) who have liked every single photo on the site (Bots & fake accounts).

Software Used: MySQL workbench 8.0 CE

Marketing

Rewarding the most loyal users:

People who have been using the platform for the longest time and I am show top 5 oldest users of Instagram.

To find the most loyal users, I am taking users table and selecting username and created_at columns. Then using the order by function, I sort the column w.r.t created_at column in ascending order. Lastly, using limit function to find top 5 oldest users of Instagram.

Query:

```
select username, created_at from ig_clone.users

order by created_at

limit 5;
```

Output/Result:

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

Remind Inactive Users to start posting:

To find the most inactive users, I first select username column from users table. Then I use left join to join photos table and lastly I find rows from the users table where the photo.id is null.

Query:

```
select users.username,users.id

from users
```

```
left join photos

on users.id= photos.user_id

where photos.id is NULL

order by users.id;
```

Output/Result:

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

Declaring Contest Winner:

The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.

To find the winner, select the column such as username, image_url, photos.id, count(*) as total. Then, I join the three table i.e. photos, users and likes tables.

Then using order by clause sort the data in descending order and lastly using limit function to view top liked photo's details.

Query:

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

Output/Result:

Id	Username	Photo_id	Image_url	Total
52	Zack_Kemmer93	145	https://jarret.name	48

So, Zack_kemmer93 is the winner of the contest and have the highest number of likes i.e. 48

Hashtag Researching:

A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform (The top 5 most commonly used hashtags on the platform).

To find the answer, first I select the columns from tags table and then join the table with photo_tags. Grouped the column by tag_name and order the value of the column and lastly using limit function to find top 5 hashtag used.

Query:

```
select tag_name,count(*) from tags

join photo_tags

on tags.id = photo_tags.tag_id

group by tag_name

order by count(*) desc

limit 5;
```

Output/Result:

Tag_name	Count(*)
smile	59
beach	42
party	39
fun	38
concert	24

Launch AD Campaign:

The team wants to know, which day would be the best day to launch ADs. (Day of the week do most users register on?)

To find the day of week first I define columns as day_of_week from dayname(created_at) column and count(*) as total_registered_users from the users table. Then, using group by function group the table on the basis of

day_of_week. And lastly, using order by function to sort the value on the basis of total registered users in descending order.

Query:

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

Output/Result:

Day_of_week	Total_registered_users
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Investor Metrics

Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds.

User Engagement:

Are users still as active and post on Instagram or they are making fewer posts. Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

To find how many times does average user posts on Instagram. First, I select user_id column and select number of rows i.e. count(*) from photos table. Then, group the table w.r.t user_id and sort the table in ascending order using user_id column.

Query:

```
select user_id,count(*) as user_post_count

from ig_clone.photos

group by user_id

order by user_id;
```

Output/Result:

User_id	User_post_count
1	5
2	4
3	4
4	3
6	5
8	4
9	4
10	3
11	5
12	4
13	5
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	12
26	5
27	1
28	4
29	8
30	2
31	1
32	4
33	5
35	2

37	1
38	2
39	1
40	1
42	3
43	5
44	4
46	4
47	5
48	1
50	3
51	5
52	5
55	1
56	1
58	8
59	10
60	2
61	1
62	2
63	4
64	5
65	5
67	3
69	1
70	1
72	5
73	1
77	6
78	5
79	1
82	2
84	2
85	2
86	9
87	4
88	11
92	3
93	2
94	1
95	2
96	3
97	2
98	1
99	3
100	2

To find total number of photos on Instagram / total number of users. First, I find the number of photos that are present in the photos.id column of the photos table. Similarly, I find the number of users that are present in the users.id column of the users table. Next, we need to divide both the values i.e. count(*) from photos/count(*) from users and hence we would get the total number of photos / total number of users.

Query:

```
select

(select count(*) from ig_clone.photos)/(select count(*) from ig_clone.users) as
avg_user_posts;
```

Result:

Avg_user_posts
2.57

Bots & Fake Accounts

The investors want to know if the platform is crowded with fake and dummy accounts. Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

To find the bots and fake accounts :

First, select the user_id column from the photos table and username column from the users table. Then, we select the count(*) function to count total number of likes from the likes table. Then I use inner join between users and likes table on the basis of users.id and likes.user_id. Then by using the group by function we group the desired output. Then, we search for the values from the count(*) from photos having equal values with the total_likes_per_user.

Query:

```
select user_id, username, count(*) as total_likes_per_user
from ig_clone.users users
inner join ig_clone.likes likes
on users.id = likes.user_id
group by likes.user_id
having total_likes_per_user = (select count(*) from ig_clone.photos)
```

Output/Result:

User id	Username	Total likes per user
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