INSTAGRAM USER ANALYTICS

DONE BY: G. REVAN

REPORT ON INSTAGRAM USER ANALYTICS:

The report is divided in two parts i.e; marketing and Investor metrics.It gives the analysis of following questions:

Marketing:-

1.Rewarding top5 loyal users

2.Remind inactive users to start posting

3.Declaring contest winners

4.Hashtag Researching

5.Best day for AD Campaign

Investor Metrics:-

1.User Engagement

2.Bots and Fake Accounts

Software used: MySQL Workbench 8.0.33 CE

PRE REQUISITES FOR THE ABOVE ANALYSIS:

1. **We need to have My SQL installed in our pc.**

2. **Create a table with the given information of Instagram users and their details.**

3. **Write the SQL statements in a structured format.**

MARKETING

The marketing team wants to reward the top 5 oldest loyal users using instagram from the database provided.

To find the most loyal users from the database we use the following steps:

1.By using select query from the users table we select username and created_at columns.

2.From above command we get all the users using Instagram.So,we need to filter the data using order_by fuction for the created_at column by giving the order_by ascending.

3.We need only the top 5 users,so we use limit function to get only 5 users.

SQL QUERY:

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

MARKETING

People(5) who have been using Instagram for the longest time.

OUTPUT/ RESULT:

username	created_at
Darby_Herzog	06-05-2016 00:14:21
Emilio_Bernier52	06-05-2016 13:04:30
Elenor88	08-05-2016 01:30:41
Nicole71	09-05-2016 17:30:22
Jordyn.Jacobson2	14-05-2016 07:56:26

MARKETING

2.Remind Inactive users to Start posting

We need to find the most inactive users who have never posted a single picture on the Instagram.

Steps to find:

1.We use select statement and from the users table we select **username** column.

2.We use join operations to join bath photos table and users table.Using left join photos table on users table(users.id=photos.user_id) we get al the users who have posted/not posted a picture.

3.Now,on the above query we add a condition(photos.id ISNULL) to get the users who have not posted a picture.

SQL QUERY:

```
select username, users.id as user_id
from users      left join photos    on users.id =
photos.user_id
where photos.id IS NULL    order by users.id;
```

MARKETING

2.Remind Inactive users to Start posting

We have a total of 26 users who have never posted a single photo on Instagram.

OUTPUT/ RESULT:

username	user_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

3.DECLARING CONTEST WINNER:

The team started a contest and the user who gets the most like on a single picture will win the contest and declare the person as winner.

Now,we need to identify the winner and provide details to the team.

Steps to find the winner:

1.First select all the columns like username , photo id ,image_url,count(*)for likes.

2.Then we inner join the three tables .

3.Now we use groupby function for diving the output on the basis of photos.id.

4.We use orderby function to sort the data in descending order(highest likes on top).

5.We use limit function to get only the user with highest likes.

MARKETING

3.DECLARING CONTEST WINNER:

The team started a contest and the user who gets the most like on a single picture will win the contest and declare the person as winner.

SQL QUERY:

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

MARKETING

3.DECLARING CONTEST WINNER:

The team started a contest and the user who gets the most like on a single picture will win the contest and declare the person as winner.

OUTPUT/ RESULT:

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

The winner of the contest is the user_id 52 and named **Zack_Kemmer93** photo_id 145 and has the highest like numbered to 48.

MARKETING

4.HASHTAG RESEARCHING:

A partner company wants to know which hashtags to use to reach to the most people on Instagram.

We need to identify the top 5 hashtags which are widely used.

Steps to find the most used hashtags:

1.From the tag table we need to select tag_name column and the count(*) as total for the number of tags used.

2.We need to join tags and photos table using tags.id=photo_tags.tag_id .

3.In the next step we need to use group_by function to get results on tags.tag_name.

4.By using order_by function we need to sort the output on total by descending order so we get the highest used tags at the top.

5.By using limit function we can get top 5 hashtags used by users on Instagram.

MARKETING

4.HASHTAG RESEARCHING:

A partner company wants to know which hashtags to use to reach to the most people on Instagram.

SQL QUERY:

```
select tags.tag_name, count(*) as
total_number_of_times_tag_used_individually
from tags
join photo_tags
on tags.id = photo_tags.tag_id
group by tags.tag_name
order by total_number_of_times_tag_used_individually DESC
limit 5;
```

MARKETING

4.HASHTAG RESEARCHING:

A partner company wants to know which hashtags to use to reach to the most people on Instagram.

OUTPUT/ RESULT:

The below table gives the number of times a hashtag is used and the "smile" hashtag has more number.

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

MARKETING

5. Launch AD Campaign

The team wants to know, which day would be the best day to launch ADs.

We need to find the day of the week on which the most users have registered on Instagram.

Steps to find the day:

1.We need to create new column for output table using dayname(created_at) as day_of_week and count(*) as total_registered from users table.

2.We use group_by function to group the table on the basis of day_of_week then we get the users registered on what days.

3.using order_by function the no.of users joined on the day_of_week in descending order we get the highest users registered on which day.

MARKETING

5. Launch AD Campaign

The team wants to know, which day would be the best day to launch ADs.

SQL QUERY:

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

MARKETING

5. Launch AD Campaign

The team wants to know, which day would be the best day to launch ADs.

OUTPUT/ RESULT :

The best day to do campaign so it could reach to the most people on Instagram is either on **THURSDAY** or **SUNDAY**.

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

INVESTOR METRICS

1. **User Engagement: Are users still as active and post on Instagram or they are making fewer posts.**

We need to find the number of times does an average user posts on instagram.

Also,We need to find the total number of photos on instagram/total number of users.

INVESTOR METRICS

Steps to find numbers of average posts made by a user:

1.So,we need o find number of photos present in the photos.id column of the photos table.We add a count(*) from photos.

2.Now,we need to find the number of users from users.id column of users tableand user count(*) to get the total number.

3.Take the average both photos and users count(*) values .

4.The value of average gives us the number of posts/number of users.

5.we need to find the total posts of each user_id from photos table to find the number of times the users posts on instagram.

INVESTOR METRICS

1. User Engagement: Are users still as active and post on Instagram or they are making fewer posts

SQL QUERY:

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

OUTPUT/ RESULT :

There are 257 rows and each row contains a photo and the ids are 100 ,so the output is 257/100.

total_photos_divide_total_photos
2.57

INVESTOR METRICS

1. **User Engagement: Are users still as active and post on Instagram or they are making fewer posts**

Now we need to find the number of times each user posts on instagram.

SQL QUERY:

```
select user_id,count(*) as user_post_count
from photos
group by user_id
order by user_id;
```

INVESTOR METRICS

1. User Engagement: Are users still as active and post on Instagram or they are making fewer posts

OUTPUT/ RESULT :

The below describes about the posts of each user.

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

INVESTOR METRICS

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

To know the bots and fake accounts we need to find the users who liked each and every picture as it not possible for a human being.

Steps to find bots and fake accounts:

1.We need to select the username,user_id and count(*) as likes_per_user from users table.

2.We need to do inner join on users table and likes.user_id table for the number of likes of each user.

3.We use group by function to find the fake accounts which liked all the photos.

INVESTOR METRICS

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

SQL QUERY:

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

INVESTOR METRICS

2. Bots & Fake Accounts: **The investors want to know if the platform is crowded with fake and dummy accounts.**

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

The user_id 's given in the table are the bots or fake accounts based on the likes posted for all other accounts.