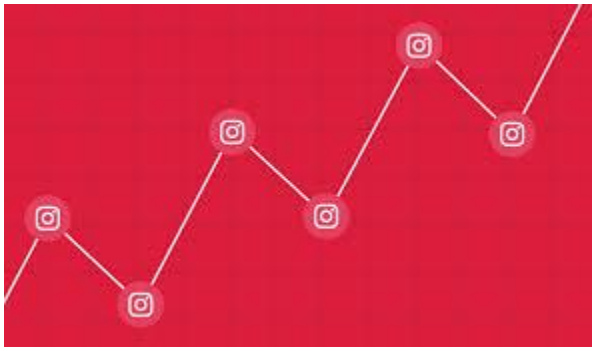
Instagram User Analytics



Project Description:

This project helps in providing useful insights from the instagram user database consisting of several tables which helps in increasing user engagement and experience on the app. Several questions like reminding or rewarding users which helps in marketing campaigns as well as questions like detecting fake accounts will also be covered.

Project Approach:

For executing the project SQL is used. A user database is created using queries with the raw data. Several data fetching queries are used to get the useful insights by providing SQL queries and the results for every question asked.

Tech Stack Used:

MySQL v5.7 is used for this project and is carried out on the website fiddle.

Insights:

A. Marketing:

1. Rewarding most loyal users.

Task: Find the top 5 oldest users of instagram.

Steps:

a. Table named users will be used.
b. Ordering the table in ascending manner by the column ‘created_at’.
c. Setting the limit to get only top 5 users.

Query SQL

```
1 -- Query
2 Select *
3 From ig_clone.users
4 Order by created_at Asc
5 Limit 5;
```

Results

Query #1 Execution time: 1ms

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22

Conclusion: Ids: 80, 67, 63 and 95 are the oldest customers.

2. Remind Inactive users to start posting.

Task: Find users who have never posted a single photo on instagram.

Steps:

a. Information of users and photos are contained in the users table and photos table.

b. Photos table has a column ‘user_id’ which has the id of the owner who posted the photo.

c. By left joining the users table with the photos table we can find those users who do not have any photo id associated with them.

Query SQL

```sql
-- Query
Select u.id,p.id as photo_id
From ig_clone.users as u
Left Join ig_clone.photos as p
On u.id=p.user_id
Where p.id Is Null;
```

Results

Query #1 **Execution time: 2ms**

id	photo_id
5	null
7	null
14	null
21	null
24	null
25	null
34	null
36	null
41	null
45	null
49	null
53	null
54	null
57	null
66	null
68	null
71	null
74	null

75	null
76	null
80	null
81	null
83	null
89	null
90	null
91	null

Conclusion: All the above users have not posted a single photo and an email reminder should be sent.

3. Declaring contest winner.

Task: Identify the user which has the most number of likes in a single photo and declare him as the contest winner.

Steps:

a. Join the photos and likes table to get the most liked photo and store the photo id in a subquery.

b. Now, using the photo id we can get the user id of it using the photos table and storing it in a subquery.

c. Getting the information of the user id using the users table.

Query SQL

```sql
-- Query
Select *
From ig_clone.users
Where id =
( Select user_id /* Subquery for getting user*/
From ig_clone.photos
Where id =
( Select p.id as photo_id /*Subquery for getting most liked photo*/
From ig_clone.photos as p
Inner Join ig_clone.likes as l
On p.id=l.photo_id
Group By p.id
Order By Count(*) Desc
Limit 1
)
)
```

Results

Query #1 **Execution time: 12ms**

id	username	created_at
52	Zack_Kemmer93	2017-01-01 05:58:22

Conclusion: Zack_kemmer93 has the most liked photo and will be the contest winner.

4. Hashtag Researching.

Task: Identify the top 5 most commonly used hashtags.

Steps:

a. Join the tags and the photo_tags table to get the name of the tags and those tags which are used under photos.
b. Grouping by the tag name and counting the occurrences of each tag.
c. Ordering in descending order and limiting the result to top 5 tags.

Query SQL

```
1 -- Query
2 Select tag_name, Count(*) as count_tags
3 From ig_clone.tags as t
4 Inner Join ig_clone.photo_tags as p
5 On t.id=p.tag_id
6 Group by tag_name
7 Order By count_tags Desc
8 Limit 5;
```

Results

Query #1 **Execution time: 1ms**

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

Conclusion: The most commonly used top 5 tags are: smile, beach, party, fun and concert.

5. Launch AD Campaign.

Task: Identify the day of the week most users registered on.

Steps:

a. Using the **DAYNAME()** function we can extract the name of the day from the given date from the created_at column in the users table.
b. Using group by we can group the day and count the occurrences of each day.
c. Then ordering the count in descending order will give us the answer.

Query SQL

```sql
-- Query
Select DAYNAME(created_at) as day,Count(*) as day_count
From ig_clone.users
Group By DAYNAME(created_at)
Order By day_count Desc;
```

Results

Query #1 Execution time: 0ms

day	day_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Conclusion: From the above result, it can be concluded that Thursday and Sunday have the most registered users. So any of them is great for ad campaigns.

B. Investor Metrics.

1. User Engagement.

Task:

1. Identify how many times a user posts on instagram.
2. Calculate total number of photos divided by total users.

Steps:

Identify how many times a user posts.

a. Using the photos table and grouping by the user id.

b. Counting the occurrences of each user id will give the number of photos posted by them.

Query SQL

```
1 -- Query
2 Select user_id,Count(*) as num_photos
3 From ig_clone.photos
4 Group By user_id
5 Order By user_id
```

Results

Query #1 Execution time: 0ms

user_id	num_photos
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

.

Steps:

Calculate total number of photos divided by total users.

a. Getting the count of all users from the users table.
b. Getting the count of all photos from the photos table.
c. Dividing them both.

Query SQL

```
1 -- Query
2 Select (
3 (select count(id) from ig_clone.photos)
4 /
5 (select count(id) from ig_clone.users)
6 ) as photos_by_users
```

Results

Query #1 **Execution time: 1ms**

photos_by_users
2.5700

Conclusion: Total number of photos divided by total number of users is 2.57.

2. Bots & Fake accounts.

Task: Provide data on users(bots) who have liked every single photo.

Steps:

a. Storing the number of photos in a subquery.
b. Joining likes and users table on user id to get the users who have liked any post.

c. Now grouping by the user id will give the number of posts liked by each user.

d. Getting only those users who match the data in the subquery.

Query SQL

```
1  -- Query
2  Select user_id as bots_id,username,Count(*) as liked_photos_per_user
3  From ig_clone.likes
4  Inner Join ig_clone.users
5  On users.id=likes.user_id
6  Group By user_id,username
7  Having Count(*) = (
8  Select Count(id) as num_photos
9  From ig_clone.photos
10 )
```

Results

Query #1 Execution time: 3ms

bots_id	username	liked_photos_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

Conclusion: In total there are 13 bots who have liked each and every photo available.