Instagram User Analytics

SQL Tasks

A)Marketing Analysis:

1)Loyal User Award

Five oldest users on Instagram are:

Results

Query #1 Execution time: 0.37ms

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

Query used to execute the above result is

Query SQL

```
1 select username, created_at
2 from ig_clone.users order by created_at asc
3 limit 5;
```

Here I initially sorted the users table in ascending order using a order by function and then took a output of top 5 from the list to derive at the desired output

2)Inactive User Engagement

Users who never posted a single photo are:

Results

Query #1 **Execution time: 0.7ms**

id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59

Results

54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

Query used to arrive at the above result is

Query SQL

```
1 select users.id,username from ig_clone.users
2 where not exists
3 ( select id from ig_clone.photos
4  where ig_clone.users.id = ig_clone.photos.user_id);
```

Here I used two queries, starting with a subquery, I extracted all the user id from photos table and used another query to negate the subquery to get the ids from users that aren't present in photos to arrive at the desired output

3)Content Winner Declaration:

The Winner of most liked photo on Instagram is

Results

Query #1 **Execution time: 2.35ms**

id	username
52	Zack_Kemmer93

The query used to execute the above result is:

Query SQL

```
1 select user.id,user.username from ig_clone.users user where user.id =(select  user_id
   from ig_clone.photos where id =(select photo_id from (select count(user_id) as
  count_like , photo_id from ig_clone.likes
2 group by photo_id order by count_like desc limit 1) as max_likes_photo) );
3
```

Here in the above query, I made use of 3 tables Likes, Photos and User. Initially used likes table to extract the photo id of the photo with highest likes and then used this id to get the id of the user from the photos table to whom the highest liked photo belongs to and finally used the user id found to arrive at the desired output using the third table users which gave the detail of the user

4) Hashtag Research:

Top 5 most used hashtags on Instagram are

Results

Query #1 **Execution time: 0.77ms**

id	tag_name
21	smile
20	beach
17	party
13	fun
18	concert

The query used to arrive at the above result is

Query SQL

```
1 with top_tags as
2
3 (select tag_id, count(*) as top_tag_id
4 from ig_clone.photo_tags group by
5 tag_id order by top_tag_id desc limit 5)
6
7 select t.id, t.tag_name from ig_clone.tags t
8 inner join top_tags tt on t.id = tt.tag_id;
9
```

As we can see, I used CTE to derive at the output

Initially I used the photo tags table to get the id of top 5 most used tags and used this table to join with tags table to derive at the desired answer of top 5 most used comments on Instagram

5)Ad Campaign Launch:

The best day of the week to launch a ad is

Results

Query #1 **Execution time: 0.98ms**

weekday	users_registered
Thursday	16
Sunday	16

The above result shows the highest number of users registered on a day of a week and the query used to execute the same is

Query SQL

```
1 with users_count as
2 (select dayname(created_at) as weekday, count(*) as users_registered
3 from ig_clone.users group by weekday order by users_registered desc)
4
5 select weekday, users_registered from users_count
6 where users_registered=(select max(users_registered) from users_count);
```

Here I used CTE and used 2 queries, initially used daytime() function and count to extract the weekdays and highest number of users registered on that by using group by and order by and since there were two days with maximum users registered I used a sub query to achieve the same using a max function

B)Investor Metrics:

1.User Engagement:

The Average Number of posts per user who posted is

average_post_per_user
3.4730

And query used to arrive at the above output is

Query SQL

```
1 select  avg(post_count)  as average_post_per_user from
2 (select user_id,count(*) as post_count from ig_clone.photos
3 group by user_id)as  post_counts;
4
5
```

In the above query I used a query with a subquery , starting with subquery I user id and post counts per user from photos table and then took the average on the post counts to arrive at the desired output

Also Total number of photos by total users on Instagram is given by

Query SQL

```
1 select
2 (select count(id) from ig_clone.photos)/(select count(id) from ig_clone.users)as
  photo_by_total_users;
3
```

Results

Query #1 **Execution time: 1.18ms**

photo_by_total_users
2.5700

2)Bots and Fake Accounts:

Bots and Fake Accounts on Instagram is given below

Results

Query #1 **Execution time: 3.07ms**

user_id	pic_count
5	257
14	257
21	257
24	257
36	257
41	257
54	257
57	257
66	257
71	257
75	257
76	257
91	257

The query used to extract the above output is

Query SQL

```
with total_photos as
(select count(id) as total_photo from ig_clone.photos)
,
user_likes as
(select user_id, count(photo_id) as pic_count from ig_clone.likes
group by user_id)

select ul.user_id, ul.pic_count from user_likes ul
join total_photos tl
on ul.pic_count=tl.total_photo;
```

Here I used a CTE containing 3 tables , starting with 1st table .

I extracted total photos from the photos table and the secondly,

I used Liked table to extract total likes of every user using a group by

and finally I used both these above tables to get the fake and bot accounts using a join operation

Project Description: Project is aimed to improve my sql skils by analysing the Instagram data provided and completing the provided tasks, the approach I took based on what I learnt from my sql classes provided in the course

Approach and Insights: Provided above after every tasks

Tech-Stack used: DB-Fiddle(MySql 8.0)

Result: This project has helped me improve my sql and problem solving skills , mainly helped me to understand problem statement and approach to take to solve that problem