Instagram User Analytics

Project Description:

User analysis is the process by which we track how users engage and interact with our digital product (software or mobile application) in an attempt to derive business insights for marketing, product & development teams.
These insights are then used by teams across the business to launch a new marketing campaign, decide on features to build for an app, track the success of the app by measuring user engagement and improve the experience altogether while helping the business grow.
You are working with the product team of Instagram and the product manager has asked you to provide insights on the questions asked by the management team.

Approach:

I loaded the dataset provided into new schema, studied it to know about the dataset then done analysis on it using SQL queries.

Tech-Stack Used:

I used online SQL Playground – DB Fiddle

Insights:

I executed the different Queries as per the given tasks and provided a detailed report. The Queries that I executed using given data and the detailed report that I filtered out is given below.

Result:

A. Marketing:

1. Rewarding Most Loyal Users: Find the 5 oldest users of the Instagram from the database provided.

Query SQL:

```
select * from ig_clone.users
order by created_at
limit 5
```

Result:

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
38	Jordyn.Jacobson2	2016-05-14 07:56:26

2. **Remind Inactive Users to Start Posting:** Find the users who have never posted a single photo on Instagram

Query SQL:

```
Select u.username
from
ig_clone.users u
left join
ig_clone.photos p
on u.id=p.user_id
where p.user_id is null
```

Result:

Query #1 Execution time: 0ms

username
Aniya_Hackett
Kasandra_Homenick
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59

Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

3. **Declaring Contest Winner:** Identify the winner of the contest and provide their details to the team

Query SQL:

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

Result:

Query #1 **Execution time: 4ms**

username	total_likes
Zack_Kemmer93	41

4. **Hashtag Researching:** Identify and suggest the top 5 most commonly used hashtags on the platform

Query SQL:

```sql
select t.tag_name,count(p.photo_id) as no_of_times_used from ig_clone.photo_tags p
inner join ig_clone.tags t on p.tag_id= t.id
group by tag_name
order by no_of_times_used desc
limit 5;
```

Result:

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

5. **Launch AD Campaign:** What day of the week do most users register on? Provide insights on when to schedule an ad campaign

Query SQL:

```
select dayname(u.created_at) as day,count(*) as total_reg
from ig_clone.users u
group by day
order by total_reg desc
limit 3
```

Result:

day	total_reg
Sunday	16
Thursday	16
Friday	15

B. Investor Metrics:

1. **User Engagement:** provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

Query SQL:

```
with CTE as (select u.id as userid,count(p.id) as photoid
from ig_clone.users u
left join ig_clone.photos p on u.id = p.user_id group by u.id)
select sum(photoid) as total_photos,
count(userid) as total_users,
sum(photoid)/count(userid) as photos_per_users from CTE
```

Result:

total_photos	total_users	photos_per_users
257	100	2.5700

2. **Bots & Fake 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).

Query SQL:

```
select u.username, l.user_id, count(*) as total_likes
from ig_clone.users u, ig_clone.likes l
where u.id=l.user_id
group by user_id
order by total_likes desc;
```

Result:

username	user_id	total_likes
Leslie67	75	257
Rocio33	21	257
Maxwell.Halvorson	24	257
Bethany20	91	257
Ollie_Ledner37	36	257
Mckenna17	41	257
Jaclyn81	14	257
Janelle.Nikolaus81	76	257
Duane60	54	257
Julien_Schmidt	57	257
Mike.Auer39	66	257
Aniya_Hackett	5	257
Nia_Haag	71	257