Instagram User Analytics

SQL Fundamentals

Description :

I assisted the attempt to generate business insights for the marketing , product and development teams in this project concerning Instagram User Analytics by sending the data to them using SQL(structured query language). During this process I am using various SQL queries to collect the needed data. By using the data I was able to determine the top-ranked oldest user,the most popular hashtag, the total no. of users , whether the user has posted any pictures or not, any may other insights about instagram.

Approach:

I first analysed the objective and looked for the actual data that the team required, and then I imported the data into SQL and ran numerous queries to understand the data and discover the insights that the team need for business benefits.

Tech Stack Used:

DB Fiddle

Version- 8.0

Results:

A) Marketing

1) Rewarding Most Loyal Users : People who have been using the platform for the longest time.
   Task – Find the 5 oldest users of the instagram from database provided.

   Query :

```
SELECT
username,
created_at
FROM
ig_clone.users
ORDER BY
created_at
LIMIT 5
```

   Output:

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

2) Remind Inactive Users to Start Posting :

By sending them promotional emails to post their first photo.

Query :

```
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
order by
u.username;
```

Output :

username
Aniya_Hackett
Bartholome.Bernhard
Bethany20
Darby_Herzog
David.Osinski47
Duane60
Esmeralda.Mraz57
Esther.Zulauf61
Franco_Keebler64
Hulda.Macejkovic
Jaclyn81
Janelle.Nikolaus81
Jessyca_West
Julien_Schmidt
Kasandra_Homenick
Leslie67
Linnea59
Maxwell.Halvorson
Mckenna17
Mike.Auer39
Morgan.Kassulke
Nia_Haag
Ollie_Ledner37
Pearl7
Rocio33
Tierra.Trantow

3) Declaring Contest Winner : The team started a contest and the user who gets the mst likes on a single photo will win the contest now they wish to declare the winner.

Task : Identify the winner of the contest and provide their details to the team.

Query :

```
with base as ( select likes.photo_id ,
               users.username ,
               count(likes.User_id) as like_User
               from ig_clone.likes likes
               inner join
               ig_clone.photos photos on
               likes.photo_id = photos.id
               inner join
               ig_clone.users users on
               photos.user_id = users.id group by
               likes.Photo_id,
               users.username order by Like_user desc limit 1 )
               select username from base;
```

Output:

username
Zack_Kemmer93

4) Hashtag Researching : A partner brand wants to know,which hashtags to use in the post to reach the most people on the platform.

Task : Identify and suggest the top 5 most commonly used hashtags on the platform.

Query:

```
SELECT t.tag_name,
COUNT(p.photo_id)AS num_tags
FROM ig_clone.photo_tags p INNER JOIN
ig_clone.tags t ON p.tag_id = t.id
GROUP BY tag_name ORDER BY
num_tags DESC LIMIT 5
```

Output:

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

5) Launch AD Campaign : The team wants to know, which day would be the best day to launch AD’s.

Task : What day of the week do most users register on ? Provide insights on when to schedule an ad campaign.

Query:

```
SELECT WEEKDAY(created_at)
AS weekday,COUNT(username)
AS num_users
FROM ig_clone.users GROUP
BY 1 ORDER BY 2 DESC
```

Output:

weekday	num_users
3	16
6	16
4	15
1	14
0	14
2	13
5	12

B) Investor metrics :

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

Task : Provide how many times does average user posts on Instagram. Also, provide the total no. of photos on Instagram total no. of users.

Query:

```
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_user FROM CTE
```

Output:

total_photos	total_users	Photos_per_user
257	100	2.5700

2. Boots and Fake Account :

The investors want to know if the platform is crowded with fake and dummy accounts.

Task : 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:

```
WITH photo_count AS (SELECT
                     user_id,COUNT(photo_id) AS num_like
                     FROM ig_clone.likes GROUP BY
                     user_id ORDER BY num_like DESC)
                     SELECT * FROM photo_count WHERE num_like =
                     (SELECT count(*) FROM ig_clone.photos)
```

Output:

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