INSTAGRAM USER ANALYTICS

Project Description

The project involves performing user analytics on the Instagram app's database using SQL and the online platform SQL Fiddle. The aim is to provide insights on user behaviour and engagement to help the marketing, product, and development teams as well as investors make informed and data-driven decisions that will help the business grow.

Approach

The first step in the project was to create a database by running the commands provided in the attachments. Next step was to connect to the database using SQL Fiddle and execute queries to find the required insights. The queries were written in SQL and executed using SQL Fiddle and the results were analysed to provide insights.

Tech-Stack Used

For this project, we used SQL Fiddle running SQL v8.0 to run queries to analyse and get insights. DB-Fiddle is an online SQL database playground for testing, debugging and sharing SQL snippets. It is an excellent platform for performing quick data analysis.

Insights

The analysis of Instagram database provided several insights which are as follows:-

1. Rewarding the most loyal users:

Query SQL

```
#Find the 5 oldest users
SELECT username, created_at
FROM ig_clone.users
Order By created_at Limit 5;

```

Using the above query 5 oldest users were found from the database and were selected to be rewarded as they are they have been using the platform for the longest time. Their details are –

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:

```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
ORDER BY u.username;
```

The above query gives us the names of all the users who haven't posted any photos on the platform ever. Below are the usernames of the users which can be used to access their email ids and be notified through email to post their first post on the platform.

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

3. Declaring the contest winner:

```sql
SELECT l.photo_id, u.username,
COUNT(l.user_id) AS like_user
FROM ig_clone.likes l
INNER JOIN ig_clone.photos p
ON l.photo_id = p.id
INNER JOIN ig_clone.users u
ON p.user_id = u.id
GROUP BY l.photo_id, u.username
ORDER BY like_user DESC LIMIT 1;
```

The marketing team started a contest in which the user with most likes on a photo wins. Running the above query provides us with the user who got the most likes on a single photo. And the winner of the contest is - Zack_Kemmer93 with 48 Likes on Photo ID – 145.

photo_id	username	like_user
145	Zack_Kemmer93	48

4. Hashtag researching:

As a partner brand wants to know the top hashtags so as to reach the maximum people on the platform. The below query provided us with the top 5 hashtags –

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

The query gave back the following results-

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

5. Launch Ad Campaign:

The marketing team wants to know which day is best to launch an ad so we ran a query to get the weekday when users logged in the most-

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

The results obtained were as follows-

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

From the results we can conclude that weekday 3 and 6 i.e., Thursday and Sunday had the most number of users log on to the platform.

6. User Engagement:

The investors want to know whether the platform is performing well or not. So, through a query we find the average number of posts by a user on Instagram and also the total number of posts and total number of users on the platform.

```sql
WITH base 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 base;
```

The above query gave us the following results-

total_photos	total_users	photos_per_user
257	100	2.5700

From the data we can infer that the platform has more photos per user and that is performing well.

7. Bots and Fake accounts:

To find out whether the platform is crowded with bots and fake accounts we ran the following query that gives us the user IDs of users who have liked all the photos on the platform which a normal user can't do.

```sql
SELECT user_id,
COUNT(*) AS countt
FROM ig_clone.likes
GROUP BY user_id
HAVING countt = (SELECT COUNT(*) FROM ig_clone.photos);
```

The following User IDs were obtained all with 257 likes each-

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

RESULTS

Through this project, we were able to provide insights on user behaviour and engagement on Instagram. These insights can be used by the marketing, product and development teams as well as investors to make informed decisions that will help the business grow. The project also demonstrated the use of SQL and SQL DB Fiddle for data analysis, which can be applied to other projects as well.