Instagram User Analytics

Sandeep Kumar Thakur

As a data analyst at Instagram, the main goal is to extract valuable insights from user data using MySQL Workbench. The insights will help the product team make informed decisions about the future direction of the Instagram app. The program addresses specific questions posed by the management team related to market research and investor criteria.

Approach:

1. Understanding the Database Schema: Begin by familiarizing myself with the database schema to understand the structure of the data and relationships between different tables.
2. Data Exploration: Conduct initial exploratory analysis to get a sense of the data distribution, identifying key metrics such as user registration dates, post counts, likes, etc.
3. Task-wise Analysis:

   A) Marketing Analysis:

   - ***Loyal User Reward:*** Identify the five oldest users based on registration dates.
   - ***Inactive User Engagement:*** Identify users who have never posted a photo.
   - ***Contest Winner Declaration:*** Determine the user with the most likes on a single photo.
   - ***Hashtag Research:*** Identify the top five most commonly used hashtags.
   - ***Ad Campaign Launch:*** Determine the day of the week with the highest user registrations for scheduling ad campaigns.

   B) Investor Metrics:

   - ***User Engagement:*** Calculate the average number of posts per user and the total posts per user to assess user activity.
   - ***Bots & Fake Accounts:*** Identify potential bots by finding users who have liked every single photo.
4. **Query Execution:** Utilize SQL queries to extract relevant data for each task, ensuring accuracy and efficiency in execution.
5. **Insights and Recommendations:** Summarize the findings for each task, providing actionable insights and recommendations based on the analysis to aid decision-making by the product and management teams.

By following this approach, the aim is to provide valuable insights that can drive strategic decisions for Instagram's growth and enhance user experience on the platform.

Insights and Results

We present the each metrics with MySQL code as Follows:

A) Marketing Analysis

1. Most loyal users

List of 5 oldest users by joining date:

To find the most loyal top 5 users on instagram we write the query as:
1. We select the **username, created_at** columns from the table users.
2. Use order by on **created_at** to order rows in ascending order.
3. Use limit to limit the output to top 5 records

Query:

```
use ig_clone;
select username, created_at
from users
order by created_at ASC
limit 5;
```

Result:

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. Inactive User Engagement:

Identify users who have never posted a single photo on Instagram. we write the query as:

1. We select **username** and **user.id** as **user_id** from **users**.
2. We left join users and photos on **user.id = photos.user_id.**
3. We output the rows which have photos.user_id as null and order by user.id.

Query:

```
use ig_clone;
select username, users.id as user_id
from users
left join photos
on users.id = photos.user_id
where photos.id IS NULL
order by users.id;
```

Result:

username	user_id
Aniya_Hackett	5
Kasandra_Homenick	7
Jaclyn81	14
Rocio33	21
Maxwell.Halvorson	24
Tierra.Trantow	25
Pearl7	34
Ollie_Ledner37	36
Mckenna17	41
David.Osinski47	45
Morgan.Kassulke	49
Linnea59	53
Duane60	54
Julien_Schmidt	57
Mike.Auer39	66
Franco_Keebler64	68
Nia_Haag	71
Hulda.Macejkovic	74
Leslie67	75
Janelle.Nikolaus81	76
Darby_Herzog	80
Esther.Zulauf61	81
Bartholome.Bernhard	83
Jessyca_West	89
Esmeralda.Mraz57	90
Bethany20	91

The output returned 26 rows.

3. Contest Winner Declaration:

Determine the winner of the contest of most likes in a single photo and provide their details to the team. We write the query as:

1. We select users.username, **photos.image_url**, and count(*) as a **total**.
2. Inner join tables photos, likes and users on **likes.photo_id = photo.id and photos.user_id = user.id.**
3. Use group by to group according to **photo.id.**
4. Use order by to group according to count of likes.
5. Use **limit** to output the top 1.

Query:

```
use ig_clone;
select
username, photos.id, photos.image_url, count(likes.user_id) as total
from photos
inner join likes
on likes.photo_id = photos.id
inner join users
on photos.user_id = users.id
group by photos.id
order by total DESC
limit 1;
```

Result:

	username	id	image_url	total
▶	Zack_Kemmer93	145	https://jarret.name	48

4. Hashtag Research:

Identify and suggest the top **five most** commonly used hashtags on the platform. We write the query as:

1. Select **tag_name** tag table and the count(*) as a **total** function so as to count the number of tags used individually.
2. Join tags table and **photo_tags** table, on **tags.id = photo_tags.tag_id** because they contain the same content in them i.e. tag_id.
3. Use group by function to group the desired output on the basis of **tags.tag_name.**
4. Use order by function to sort the output on the basis of total in descending order.
5. Use **limit** to return the **top 5 rows** from the output.

Query:

```
use ig_clone;
select tags.tag_name,
count(*) as total
from photo_tags
join tags
on photo_tags.tag_id = tags.id
group by tags.id
order by total DESC
limit 5;
```

Result:

	tag_name	total
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

5. Ad Campaign Launch:

Determine the day of the week when most users register on Instagram.
Provide insights on when to schedule an ad campaign.We write the query as:

1. Select dayname(created_at) as **day** and count(*) as **total** from the users table.
2. then group the output table on the basis of **day**.
3. order the functions according to **total** in descending order.

Query:

```
use ig_clone;
select dayname(created_at) as day, count(*) as total
from users
group by day
order by total DESC
limit 1;
```

Result:

	day	total
▶	Thursday	16

B) Investor Metrics:

1. **User Engagement:** Calculate the average number of posts per user and the total posts per user to assess user activity.

   Query:

```
use ig_clone;
select
(select count(*) from photos) / (select count(*) from users) as avg;
```

   Result:

	avg
▶	2.5700

2. **Bots & Fake Accounts:** Identify potential bots by finding users who have liked every single photo.

   Query:

```
use ig_clone;
select user_id, username, count(*) as num_likes
from users u
join likes l on u.id = l.user_id
group by u.id
having num_likes = (select count(*) from photos);
```

   Result:

user_id	username	num_likes
5	Aniya_Hackett	257
14	Jadyn81	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