Instagram User Analytics

Project Description: User Engagement and Interaction Analysis for Instagram

As a data analyst at Instagram, this project focuses on analysing user interactions and engagement with the Instagram app to generate actionable insights for business growth. By examining how users engage with key features of the app, the analysis provides valuable data that informs strategic decisions across multiple teams. These insights help the marketing team optimize campaigns, guide the product team in developing new features, and support the development team in enhancing the overall user experience. The ultimate goal is to leverage data to improve user satisfaction, drive engagement, and support the continuous growth of the platform

Approach:
Collaborated with the product team to define key business questions and metrics. Analysed Instagram user data to identify relationships and trends. Extracted and cleaned relevant user interaction and engagement data from internal databases. Conducted targeted analyses to provide actionable insights, including:

1. Identifying loyal users with long-term engagement.
2. Highlighting users with minimal interactions for re-engagement opportunities.
3. Recognizing contest winners based on the highest likes on a single photo.
4. Conducting hashtag research to identify the most commonly used hashtags.
5. Determining the optimal day for campaign launches.

Delivered findings to the product and marketing teams, enabling data-driven decision-making to improve user engagement and drive growth.

Tech-Stack Used:

MySQL Workbench
Version 8.0.31 build 2235049 CE (64 bits)

Saloni Mate

Insights:

1. **User Loyalty:** Identified the most loyal users with the longest app usage, providing insights into retention strategies.
2. **Inactive Users:** Flagged users with minimal or no activity, enabling re-engagement campaigns.
3. **Contest Winners:** Determined users with the highest likes on a single post, valuable for promoting contests and rewards programs.
4. **Popular Hashtags:** Revealed the most frequently used hashtags, aiding content creation and marketing strategies.
5. **Optimal Ad Launch Day:** Identified the best day of the week for ad campaigns, maximizing visibility and engagement.

User Engagement:

- Analysed posts per user and determined the average number of posts, providing a benchmark for user activity.

Bots and Fake Accounts:

- Detected patterns indicative of bot or fake accounts, enhancing data integrity and platform security.

These insights provided actionable strategies to improve user retention, engagement, and marketing effectiveness.

Achievements:

Through this project, I successfully delivered actionable insights that helped the product and marketing teams improve user retention, engagement, and campaign effectiveness.

This analysis enabled Instagram to make data-driven decisions, optimize user experience, and target growth opportunities effectively. Personally, the project enhanced my skills in user behaviour analysis, while reinforcing the importance of leveraging data to drive business outcomes.

Results:

A) Marketing Analysis:

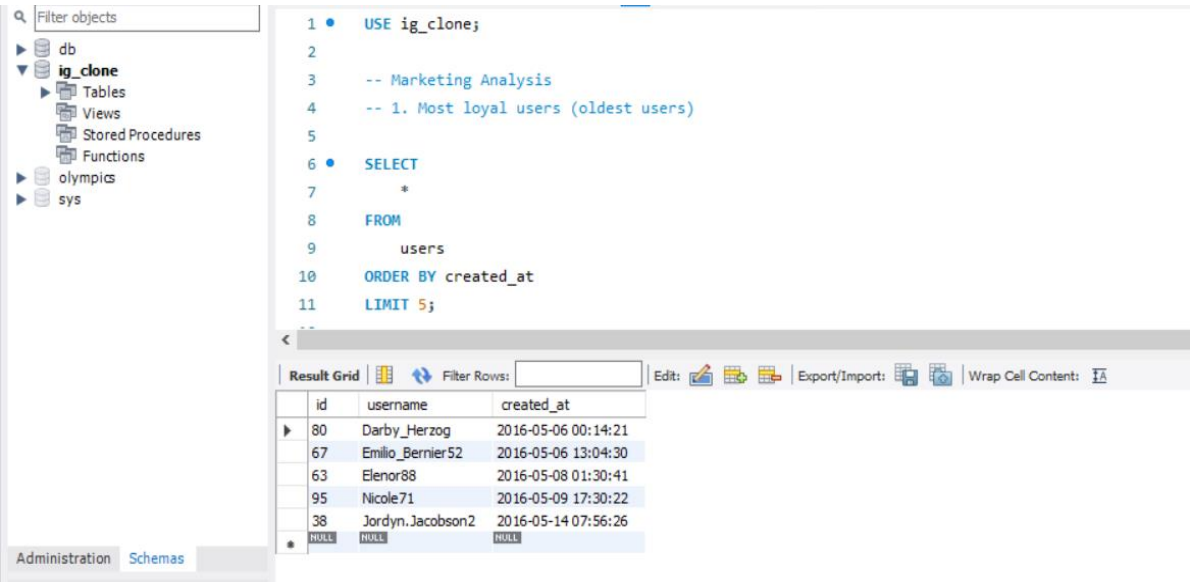
1. Loyal User Reward:

Users who have been using the platform for the longest time.

1. Darby_Herzog
2. Emilio_Bernier52
3. Elenor88
4. Nicole71
5. Jordyn.Jacobson2

SQL Query:

```
SELECT
    *
FROM
    users
ORDER BY created_at
LIMIT 5;
```

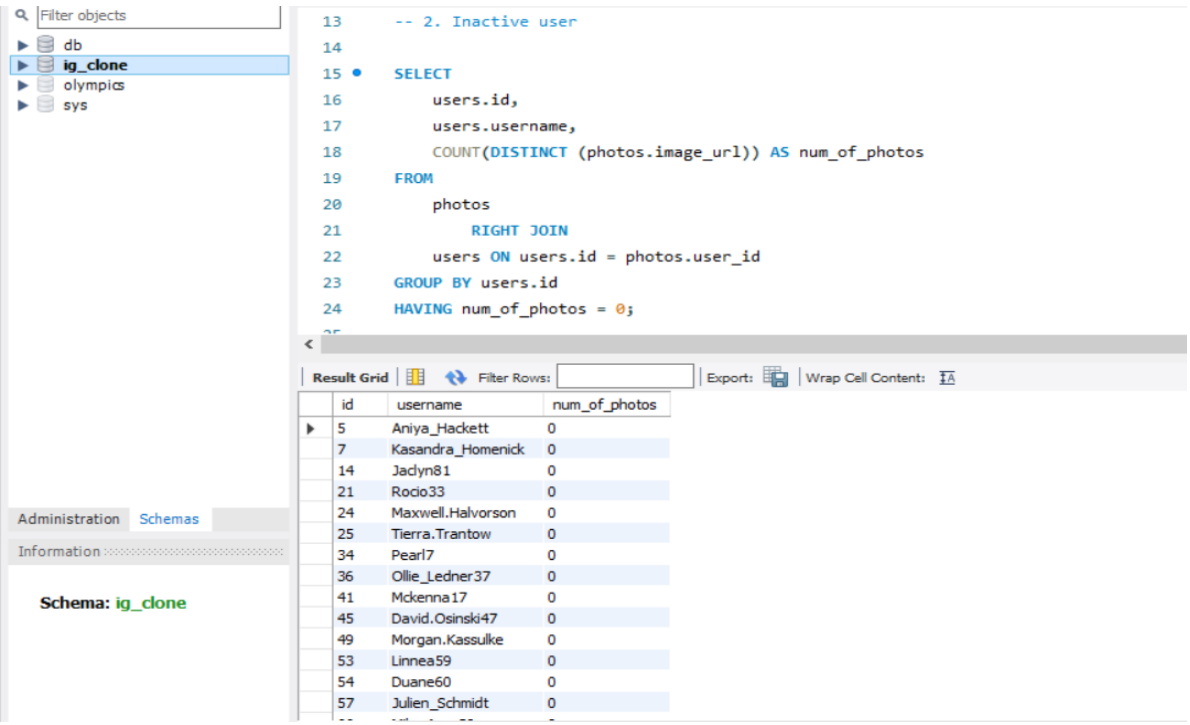


2. Inactive User Engagement

Users who have never posted a single photo on Instagram.

SQL Query:

```
SELECT
   users.id,
   users.username,
   COUNT(DISTINCT (photos.image_url)) AS num_of_photos
FROM
   photos
      RIGHT JOIN
   users ON users.id = photos.user_id
GROUP BY users.id
HAVING num_of_photos = 0;
```

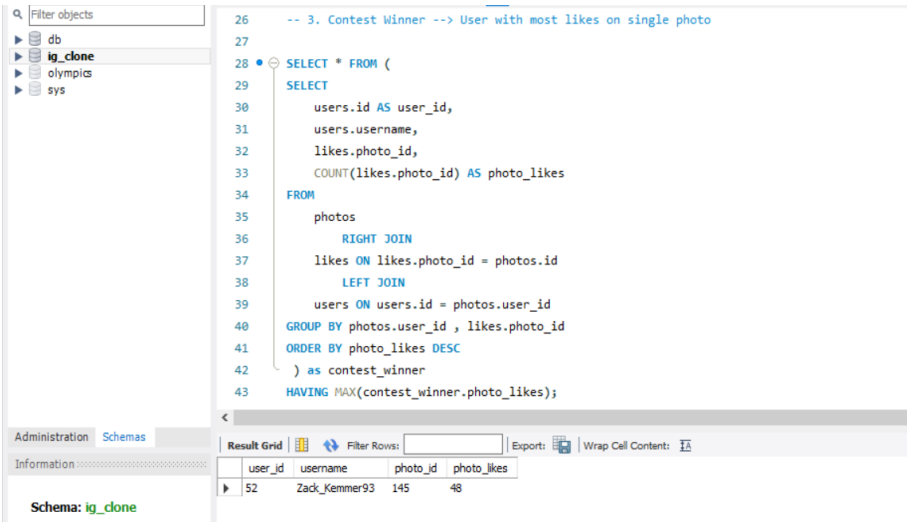


3. Contest Winner Declaration:

Winner of the contest - User with the most likes on a single photo

SQL Query:

```
SELECT * FROM (
SELECT
   users.id AS user_id,
   users.username,
   likes.photo_id,
   COUNT(likes.photo_id) AS photo_likes
FROM
   photos
      RIGHT JOIN
   likes ON likes.photo_id = photos.id
      LEFT JOIN
   users ON users.id = photos.user_id
GROUP BY photos.user_id , likes.photo_id
ORDER BY photo_likes DESC
 ) as contest_winner
HAVING MAX(contest_winner.photo_likes);
```

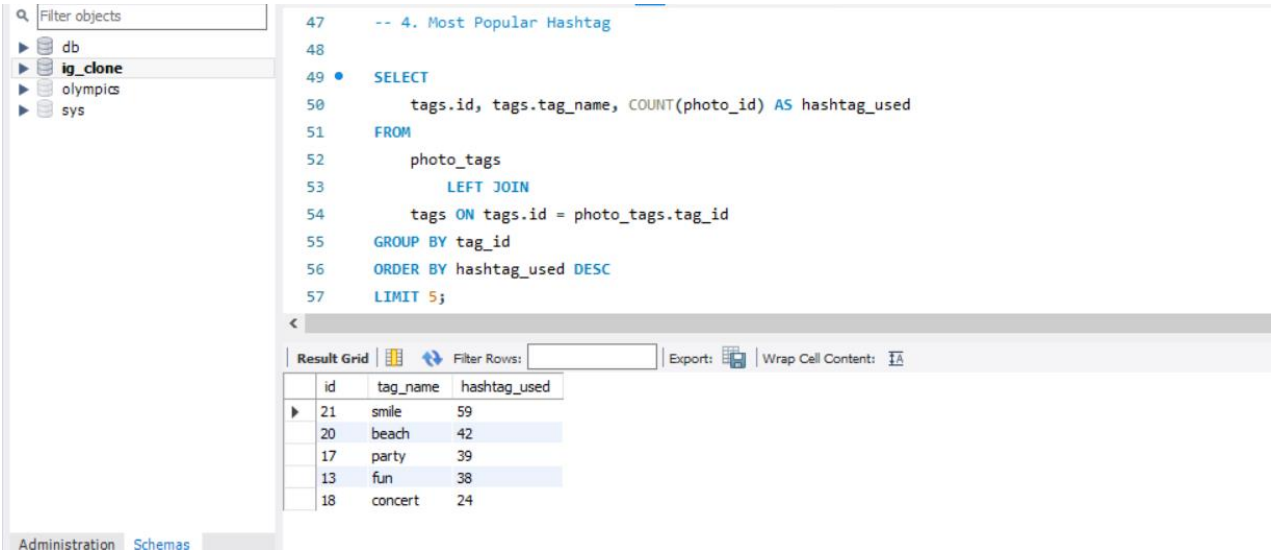


4. Hashtag Research:

Most popular hashtags: smile, beach, party, fun, concert

SQL Query:

```
SELECT
   tags.id, tags.tag_name, COUNT(photo_id) AS hashtag_used
FROM
   photo_tags
      LEFT JOIN
   tags ON tags.id = photo_tags.tag_id
GROUP BY tag_id
ORDER BY hashtag_used DESC
LIMIT 5;
```

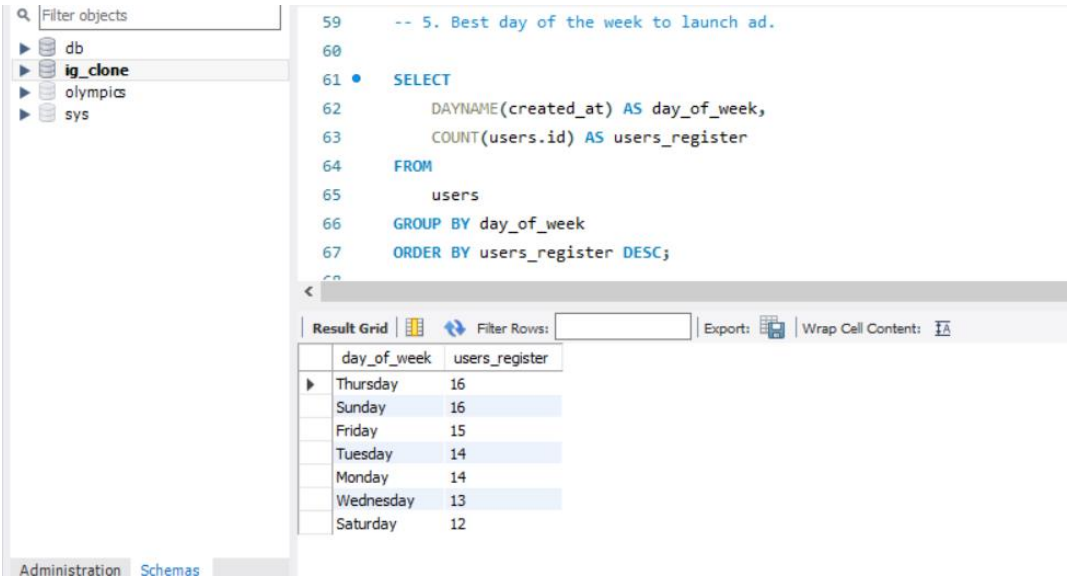


5. Ad Campaign Launch

The best day of the week to launch ads: Thursday and Sunday

SQL Query:

```
SELECT
    DAYNAME(created_at) AS day_of_week,
    COUNT(users.id) AS users_register
FROM
    users
GROUP BY day_of_week
ORDER BY users_register DESC;
```



B) Investor Metrics:

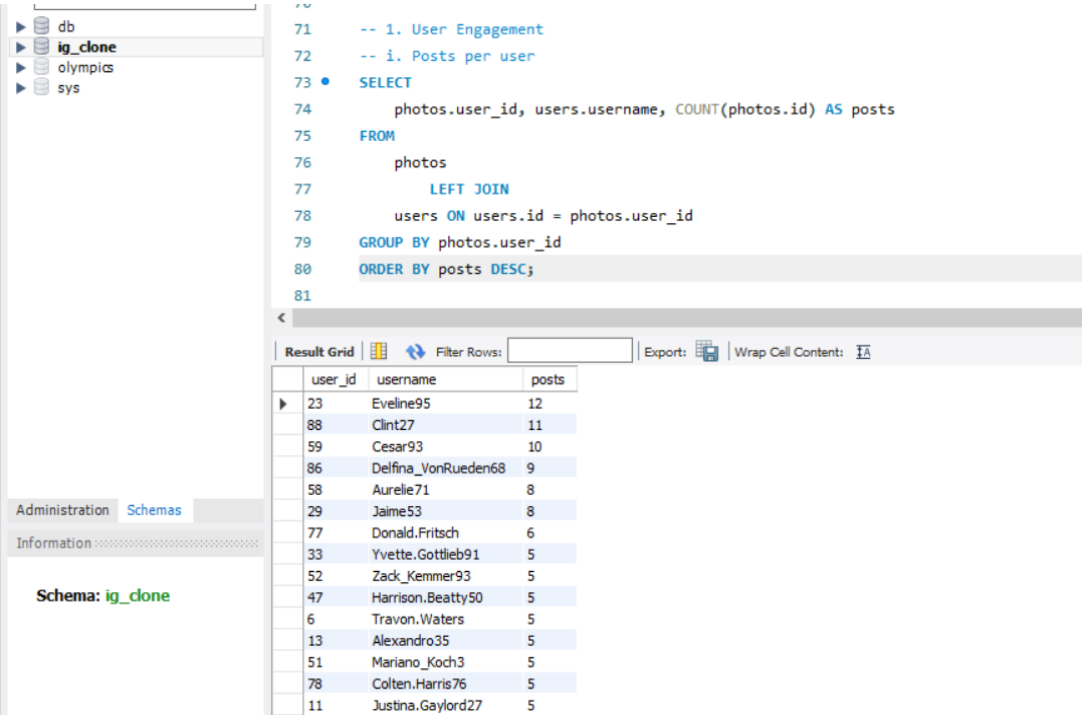
1. User Engagement:

User posts fewer posts but they are active on Instagram

SQL Query:

- Posts per user

```
SELECT
   photos.user_id, users.username, COUNT(photos.id) AS posts
FROM
   photos
      LEFT JOIN
   users ON users.id = photos.user_id
GROUP BY photos.user_id
ORDER BY posts DESC;
```

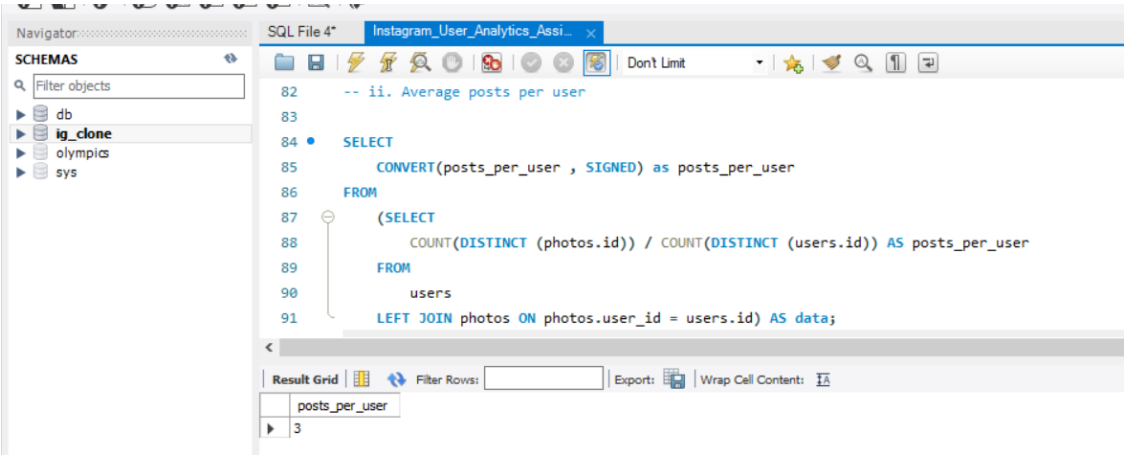


user_id	username	posts
23	Eveline95	12
88	Clint27	11
59	Cesar93	10
86	Delfina_VonRueden68	9
58	Aurelie71	8
29	Jaime53	8
77	Donald.Fritsch	6
33	Yvette.Gottlieb91	5
52	Zack_Kemmer93	5
47	Harrison.Beatty50	5
6	Travon.Waters	5
13	Alexandro35	5
51	Mariano_Koch3	5
78	Colten.Harris76	5
11	Justina.Gaylord27	5

- Average posts per user

SQL Query:

```
SELECT
   CONVERT(posts_per_user , SIGNED) as posts_per_user
FROM
   (SELECT
      COUNT(DISTINCT (photos.id)) / COUNT(DISTINCT (users.id)) AS posts_per_user
   FROM
      users
   LEFT JOIN photos ON photos.user_id = users.id) AS data;
```



2. Bots & Fake Accounts

There are 13 potential Bots and Fake accounts on the platform

SQL Query:

```
SELECT
   *
FROM
   (SELECT
      users.id AS user_id,
                              users.username,
                              COUNT(DISTINCT (likes.photo_id)) AS likes_by_user
   FROM
      users
   RIGHT JOIN likes ON likes.user_id = users.id
   GROUP BY users.id , users.username
   ORDER BY likes_by_user DESC) AS bot_detector
HAVING likes_by_user = (SELECT
      COUNT(photos.id)
   FROM
      photos)
ORDER BY bot_detector.user_id;
```

```sql
-- 2. Bots or Fake accounts

SELECT
    *
FROM
    (SELECT
        users.id AS user_id,
        users.username,
        COUNT(DISTINCT (likes.photo_id)) AS likes_by_user
    FROM
        users
    RIGHT JOIN likes ON likes.user_id = users.id
    GROUP BY users.id , users.username
    ORDER BY likes_by_user DESC) AS bot_detector
HAVING likes_by_user = (SELECT
        COUNT(photos.id)
    FROM
        photos)
ORDER BY bot_detector.user_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	username	likes_by_user
5	Aniya_Hackett	257
14	Jadyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257