INSTAGRAM USER ANALYTICS

- Jinanshi Shah

PROJECT DESCRIPTION:

The project's core objective is to employ data analytics to furnish strategic insights for stakeholders in an dummy Instagram. Key tasks involve recognizing devoted users, re-engaging inactive ones, declaring winners in contests, conducting hashtag research, pinpointing optimal days for ad campaign launches, and evaluating both user engagement and the potential presence of automated entities. By utilizing SQL queries, the analysis aims to guide decision-making processes, refine marketing strategies, and advance the overall performance of the platform. The emphasis is placed on deriving actionable insights from user behavior data, ensuring a technology-driven approach that facilitates informed decision-making and continual platform enhancement.

APPROACH :

- Understanding the Database Schema:
  Review the provided SQL database schema to understand the relationships between tables and the data available.
- Data Exploration:
  Use SQL queries to explore the data in each table (e.g., users, photos, likes) to gain insights into the available information.
- Task-Specific Queries:
  Develop SQL queries tailored to each task:
- For identifying loyal users: Use the users table and order by registration date.
- For engaging inactive users: Identify users who have never posted a photo by comparing user IDs between the users and photos tables.
- For declaring contest winners: Determine users with the most likes on a single photo by querying the likes and photos tables.
- For hashtag research: Identify the top five hashtags based on usage counts from the tags and photo_tags tables.
- For ad campaign launch day: Analyze user registration days from the users table.
- For evaluating user engagement: Calculate the average number of posts per user and the photos-to-users ratio using data from the photos and users tables.
- For potential bots or fake accounts: Identify users who have liked every single photo, suggesting potential automated behavior.
- Execution and Analysis:
  Execute the SQL queries against the database to retrieve relevant information.
  Analyze the results to derive insights, such as trends, patterns, and anomalies.
- Documentation:
  Document the findings, insights, and any recommendations in a comprehensive report.

TECH – STACK USED :

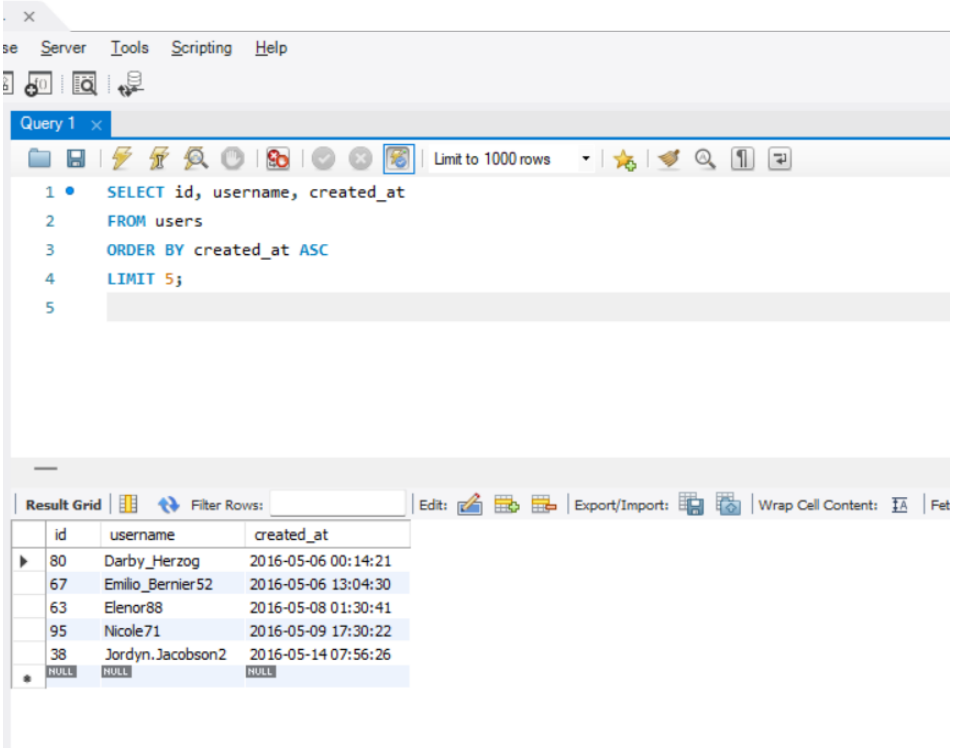
- MySQL WorkBench – User Optimised S/w for demonstrating SQL Queries.
- SQL Server
- SQL Shell

SQL Tasks :

A) Marketing Analysis:

1. **Loyal User Reward:**

   - Query & Output:



2. **Inactive User Engagement:**

- Query:

```
SELECT id, username
FROM users
WHERE id NOT IN (SELECT DISTINCT user_id FROM photos);
```

- Output:

id	username
14	Jadyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

3. **Contest Winner Declaration:**

- Query:

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

- Output:

user_id	username	photo_id	like_count
52	Zack_Kemmer93	145	48

#	Time	Action	Message
22	13:16:54	SELECT id, username FROM users WHERE id NOT IN (SELECT DISTINCT user_id FROM photos) LIMIT 0, 1...	26 row(s) returned
23	13:18:45	SELECT u.id AS user_id, u.username, p.id AS photo_id, COUNT(l.user_id) AS like_count FROM users u JOIN ...	1 row(s) returned

4. **Hashtag Research:**

- Query & Output:

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

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

5. **Ad Campaign Launch:**

- Query:

```
SELECT DAYNAME(created_at) AS registration_day, COUNT(id) AS registration_count
FROM users
GROUP BY registration_day
ORDER BY registration_count DESC;
```

- Output:

registration_day	registration_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

#	Time	Action	Message
24	13:20:09	SELECT t.tag_name, COUNT(pt.photo_id) AS usage_count FROM tags t JOIN photo_tags pt ON t.id = pt.tag_...	5 row(s) returned
25	13:20:54	SELECT DAYNAME(created_at) AS registration_day, COUNT(id) AS registration_count FROM users GROUP ...	7 row(s) returned

B) Investor Metrics:

1. **User Engagement:**

   - Query & Output :

```
SELECT AVG(posts_per_user) AS average_posts_per_user
FROM (
    SELECT user_id, COUNT(*) AS posts_per_user
    FROM photos
    GROUP BY user_id
) AS user_posts;
```

average_posts_per_user
3.4730

#	Time	Action	Message	Duration / Fetch
25	13:20:54	SELECT DAYNAME(created_at) AS registration_day, COUNT(id) AS registration_count FROM users GROUP ...	7 row(s) returned	0.015 sec / 0.000 sec
26	13:23:02	SELECT AVG(posts_per_user) AS average_posts_per_user FROM (SELECT user_id, COUNT(*) AS posts_...	1 row(s) returned	0.000 sec / 0.000 sec

   - Query & Output :

```
SELECT (SELECT COUNT(*) FROM photos) AS total_photos,
       (SELECT COUNT(*) FROM users) AS total_users,
       (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS photos_per_user_ratio;
```

total_photos	total_users	photos_per_user_ratio
257	100	2.5700

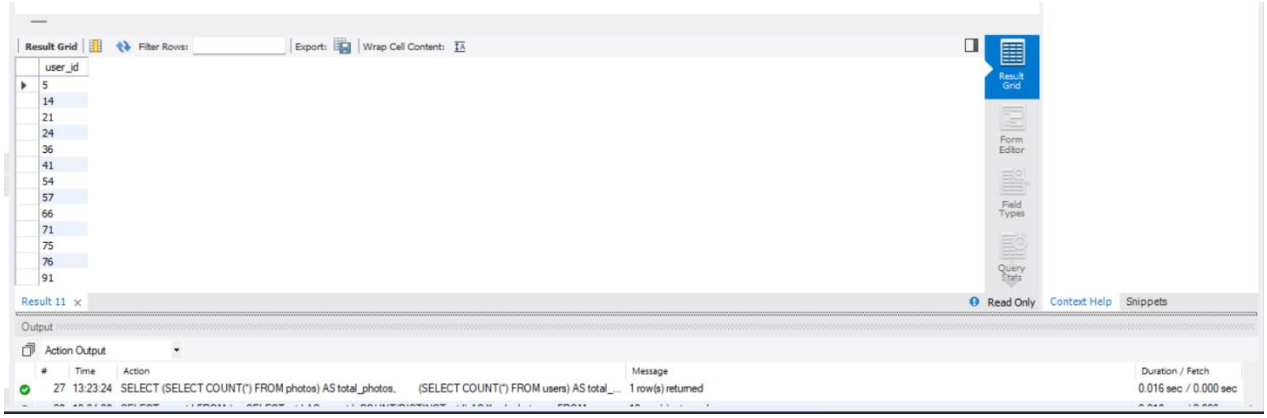
#	Time	Action	Message	Duration / Fetch
26	13:23:02	SELECT AVG(posts_per_user) AS average_posts_per_user FROM (SELECT user_id, COUNT(*) AS posts_...	1 row(s) returned	0.000 sec / 0.000 sec
27	13:23:24	SELECT (SELECT COUNT(*) FROM photos) AS total_photos, (SELECT COUNT(*) FROM users) AS total_...	1 row(s) returned	0.016 sec / 0.000 sec

2. **Bots & Fake Accounts:**

- Query:

```
SELECT user_id
FROM (
    SELECT u.id AS user_id, COUNT(DISTINCT p.id) AS liked_photos
    FROM users u
    JOIN likes l ON u.id = l.user_id
    JOIN photos p ON l.photo_id = p.id
    GROUP BY u.id
) AS user_likes
WHERE liked_photos = (SELECT COUNT(*) FROM photos);
```

- Output:



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

#	Time	Action	Message	Duration / Fetch
27	13:23:24	SELECT (SELECT COUNT(*) FROM photos) AS total_photos, (SELECT COUNT(*) FROM users) AS total_...	1 row(s) returned	0.016 sec / 0.000 sec

INSIGHTS:

- Loyal Users:
  Identified five oldest users, indicating long-term loyalty.
- Inactive User Engagement:
  Isolated users with zero posts, suggesting a need for targeted engagement.
- Contest Winner:
  Determined winner based on the most likes on a single photo.
- Hashtag Research:
  Revealed top five hashtags for effective content promotion.
- Ad Campaign Launch:
  Identified peak user registration day for optimal ad launches.
- User Engagement:
  Calculated average posts per user for assessing overall engagement.
- Bots & Fake Accounts:
  Detected potential bots through users liking every photo

RESULT :

Participating in this project enriched my understanding of user dynamics and strategic analysis. Recognizing loyal users underscored the importance of effective community-building, while re-engaging dormant users highlighted the impact of personalized approaches. Analyzing ad campaign optimization elucidated the nuanced relationship between timing and marketing effectiveness. Proactively addressing potential automated behaviour emphasized the critical role of maintaining platform integrity and user trust. Overall, this experience significantly honed my analytical skills, accentuating the pivotal importance of data-driven insights in shaping strategic decisions and elevating user experiences within the continually evolving digital landscape.