Instagram User Analytics

Overview:

In this project, I will be using SQL and MySQL Workbench as your tool to analyze Instagram user data and answer questions posed by the management team. My insights will help the product manager and the rest of the team make informed decisions about the future direction of the Instagram app.

the goal of this project is to use your SQL skills to extract meaningful insights from the data. this will help solve some of the important queries that can provide meaningful information and can make the company's handling task easier and can make the application more productive.

Approach:

I used My SQL workbench 8.0 CE to execute my queries and fetch my data. First I created a database name ig_clone and then created my tables such as users, photos, tags, liked and many other tables containing all the necessary constraints. Then I inserted my data into the respective tables. then I used my SQL queries and various functions such as joins, group by clause, aggregate functions, nested sub queries to fetch my data.

Tech-Stack:

I will be using My SQL workbench 8.0 CE to complete my project.

I choose this application because it is one of the best software used in analysis of data as it is fast, robust, have data security.it is free from data redundancy and data inconsistency and also it holds ACID properties.

It provides a lot of functions and modulations such ass aggregate functions, window functions, subqueries, nested subqueries and correlated subqueries which makes data fetching easier and helps us to perform our task conveniently.

Insights and results:

During the course of this project I got deep knowledge of SQL uses and learnt how to use it in a huge database management system which in this case was analysis of Instagram data. I fetched most active users, most like photos, most used tags and many more data during the course of this project and used my knowledge of SQL to do my task.it was a very good experience putting knowledge into use. I learned a lot in this project and deepen my SQL skills.

Part A:

Query 1:

```
select username,created_at as oldest_users from users order by created_at asc limit 5;
```

Output:

username	oldest_users
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

Query 2:

```
SELECT
  users.username
FROM
  users
  LEFT JOIN photos ON users.id = photos.user_id
WHERE
  photos.id IS NULL;
```

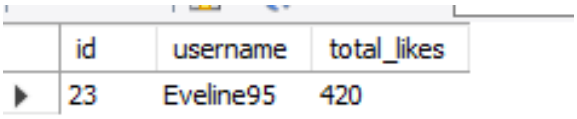
Output:

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

Query 3:

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

Output:



id	username	total_likes
23	Eveline95	420

Query 4:

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

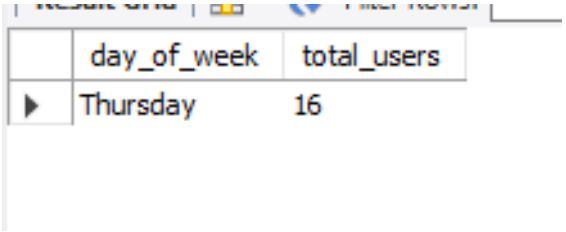
Output:

tag_name	count_total_tag
smile	59
beach	42
party	39
fun	38
food	24

Query 5:

```sql
SELECT
    DAYNAME(created_at) AS day_of_week, COUNT(*) AS total_users
FROM
    users
GROUP BY day_of_week
ORDER BY total_users DESC
LIMIT 1;
```

Output:



	day_of_week	total_users
▶	Thursday	16

Part B:

Query 1:

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

Output:

average_posts_per_user	total_photos	total_users
2.5700	257	100

Query 2:

```
SELECT
  user_id  from likes
GROUP BY
  user_id
HAVING
  COUNT(DISTINCT photo_id) = (
    SELECT
      COUNT(*)
    FROM
      photos
  )
```

Output:

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