Task-2: Instagram User Analytics

SQL Tasks:

- **Part(A) - Analysis For Marketing Team:**

1. Rewarding the Loyal Users
2. Encouraging Inactive Users to Start Posting
3. Announcing the Contest Winners
4. Conducting Hashtag Research
5. Launching Ad Campaign

- **Part(B) - Metrics For Investors:**

1. User Engagement
2. Identifying Bots and Fake Accounts

- **Software Used: MySQL Workbench 8.0 CE**

MARKETING ANALYSIS

1) Rewarding the loyal users

Team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.
Task: Identify the five oldest users on Instagram from the provided database.

- To find the top 5 oldest user accounts, we first use the **users table** and **select usernames** and **created_at** columns.
- Then by using **order by** function we will sort the **created_at** column to get the desired output.
- Then we use the **limit** function to get the Top 5 oldest user accounts.

Query:

```
select * from users order by created_at limit 5;
```

select * from users
order by created_at limit 5;

MARKETING ANALYSIS

1) Rewarding the loyal users

Team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.
Task: Identify the five oldest users on Instagram from the provided database.

Result/Output:

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26

MARKETING ANALYSIS

2) Encouraging Inactive Users to Start Posting

The team wants to encourage inactive users to start posting by sending them promotional emails.
Task: Identify users who have never posted a single photo on Instagram.

- Firstly to find the inactive users, we will **select username** column from the **users** table.
- Then we use **left join** function to join the **photos** table and the **users** table.
- We used **on users.id=photos.user_id** as both **users.id and photos.user_id** have some common contents
- Finally, we will find the rows from the **users** table **where the photos.id is null**.

Query:

```
select username,users.id as username from users left join photos on users.id=photos.user_id where photos.id is null;
```

select username, users.id as username from users
left join photos on users.id=photos.user_id
where photos.id is null;

MARKETING ANALYSIS

2) Encouraging Inactive Users to Start Posting

Result/Output:

There are **26 users** out of **100 users** who never posted a single photo on Instagram.

Aniya_Hackett	5
Kasandra_Homenick	7
Jadyn81	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

MARKETING ANALYSIS

3) Announcing the contest winners

The team has organized a contest where the user with the most likes on a single photo wins.

Task: Determine the winner of the contest and provide their details to the team.

- Firstly, we will **select** the **usersname** from the **users**, **photos.id**, **photos.image_url** and **count(*) as total**.
- Then, we will **inner join** the three tables i.e **photos**, **likes** and **users** on **likes.photo_id=phots.id** and **photos.user_id = users.id.**
- Then by using **group by** function we will group the results on the basis of **photos.id**
- Then, using **order by** function we will sort the data on the basis of the **total** in **descending order**.
- To find the most liked photo we will use the **limit** function to view only the most liked photo's information.

MARKETING ANALYSIS

3) Announcing the contest winners

The team has organized a contest where the user with the most likes on a single photo wins.

Task: Determine the winner of the contest and provide their details to the team.

Query:

```
select users.id as user_id,users.username, photos.id as pic_id, photos.image_url, count(*) 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;
```

select users.id as user_id, users.username, photos.id as pic_id, photos.image_url, count(*) 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;

MARKETING ANALYSIS

3) Announcing the contest winners

The team has organized a contest where the user with the most likes on a single photo wins.

Task: Determine the winner of the contest and provide their details to the team.

Result/Output:

Result Grid | Filter Rows: | Export:

user_id	username	pic_id	image_url	total
52	Zack_Kemmer93	145	https://jarret.name	48

Hence, the user named **Zack_Kemmer93** with **user_id 52** is the winner of this contest with highest number of likes i.e **48** on his photo **pic_id 145**.

MARKETING ANALYSIS

4) Conducting Hashtag Research

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
Task: Identify and suggest the top five most commonly used hashtags on the platform.

- To find the top 5 most commonly used Hastags, we need to **select** the **tag_name** column from the **tag table** and **count(*) as total_tags_used**
- Then, we need to **join tags** table and **photo_tags** table using **on tags.id = photo_tags.tag_id** as they contain the same contents in them.
- Then using the group function, we need to **group** the desired results on the basis of **tags.tag_name**
- Then using the **order by** function we need to sort the results on the basis of **total_tags_used** in descending order using **DESC**.
- Hence, to obtain the top 5 most used tag names we will use the **limit 5** function.

MARKETING ANALYSIS

4) Conducting Hashtag Research

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
Task: Identify and suggest the top five most commonly used hashtags on the platform.

Query:

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

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

MARKETING ANALYSIS

4) Conducting Hashtag Research

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.
Task: Identify and suggest the top five most commonly used hashtags on the platform.

Result/Output:

Hence, the top 5 most used Hastags are **smile**, **beach**, **party**, **fun** and **concert**.

esult Grid | Filter Rows:

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

MARKETING ANALYSIS

5) Launching AD Campaign

The team wants to know the best day of the week to launch ads.
Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

- To find the day in which most users register, we define the columns of the desired results **table**.
- using **select dayname(created_at) as day_of_week** and **count(*) as most_users_registered** from the **users** table.
- Then we use the **group by** function to group the result table on the basis of the **day_of_week**
- Using the **order by** function we sort the results table on the basis of **most_users_registered** in descending order by **DESC.**

MARKETING ANALYSIS

5) Launching AD Campaign

The team wants to know the best day of the week to launch ads.
Task: Determine the day of the week when most users register on Instagram.
Provide insights on when to schedule an ad campaign.

Query:

```
select dayname(created_at) as day_of_week, count(*) as most_users_registered from users group by day_of_week order by most_users_registered DESC;
```

select dayname(created_at) as day_of_week, count(*) as most_users_registered from users group by day_of_week order by most_users_registered DESC;

MARKETING ANALYSIS

5) Launching AD Campaign

The team wants to know the best day of the week to launch ads.
Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

Result/Output:

So, the most users are registered on **Thursday** and **Sunday**. Hence, these are the best days to launch an Ad Campaign.

Result Grid | Filter Rows:

day_of_week	most_users_registered
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

INVESTOR METRICS

1) User Engagement

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
Task: Calculate the average number of posts per user on Instagram.
Also, provide the total number of photos on Instagram divided by the total number of users.

- Firstly, we need to find how many times does average user posts on Instagram.
- To achieve that we first need to count the number photos(posts) that are present in the **photos.id** column of the **photos table** by using **count(*) from photos**
- Similarly, we need to find the number of users that are present in the **users.id** column of the **users table** using **count(*) from users**.
- Next step is to divide both values i.e **count(*) from users photos/ count(*) from users** and hence we get the total number of photos/total number of users
- To find how many times the users posts on Instagram we need to find the total of each **user_id** in **photos** table.

INVESTOR METRICS

1) User Engagement

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
Task: Calculate the average number of posts per user on Instagram.
Also, provide the total number of photos on Instagram divided by the total number of users.

Query:

```
select(select count(*) from photos)/(select count(*) from users) as avg_engage;
```

Select(select count(*) from photos)/select count(*) from users as avg_engage;

INVESTOR METRICS

1) User Engagement

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
Task: Calculate the average number of posts per user on Instagram.
Also, provide the total number of photos on Instagram divided by the total number of users.

Result/Output:

So, there are **257 rows** in photos table and **100 rows** in users table.
Hence, 257/100 = 2.5700.

Result Grid

	avg_engage
▶	2.5700

INVESTOR METRICS

1) User Engagement

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
Task: Calculate the average number of posts per user on Instagram.
Also, provide the total number of photos on Instagram divided by the total number of users.

Query to find the time each user posts on instagram:

```
select user_id,count(*) as post_count from photos group by user_id order by user_id;
```

Select user_id, count(*) as post_count from photos group by user_id order by user_id;

INVESTOR METRICS

1) User Engagement

Result/Output: **user_id** along with **user_post_count** is provided.

user_id	user_post_count
1	5
2	4
3	4
4	3
6	5
8	4
9	4
10	3
11	5
12	4
13	5
15	4
16	4
17	3
18	1
19	2
20	1
22	1
23	12
26	5
27	1
28	4

user_id	user_post_count
29	8
30	2
31	1
32	4
33	5
35	2
37	1
38	2
39	1
40	1
42	3
43	5
44	4
46	4
47	5
48	1
50	3
51	5
52	5
55	1
56	1
58	8

user_id	user_post_count
59	10
60	2
61	1
62	2
63	4
64	5
65	5
67	3
69	1
70	1
72	5
73	1
77	6
78	5
79	1
82	2
84	2
85	2
86	9
87	4
88	11
92	3

user_id	user_post_count
93	2
94	1
95	2
96	3
97	2
98	1
99	3
100	2

INVESTOR METRICS

2) Identifying Bots and Fake Accounts

Investors want to know if the platform is crowded with fake and dummy accounts. **Task:** Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

- Firstly, to Identify the bots and fake accounts, we **select** the **user_id** and **username** column **from** the **users** table.
- Then, we **select** the **count(*)** function to count total number of likes from the **likes** table.
- Then, using **inner join users** and **likes** table on the basis of **user.id** and **likes.user_id**, using the **on** function.
- Using the **group b**y function we group the desired results table on the basis of **likes.user_id**
- Then, we search for the values from the **count(*) from photos** having equal values with **total_likes_per_user**

INVESTOR METRICS

2) Identifying Bots and Fake Accounts

Investors want to know if the platform is crowded with fake and dummy accounts.
Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

Query:

```
select user_id, username, count(*) as total_likes_per_user from users inner join likes on
users.id = likes.user_id group by likes.user_id having total_likes_per_user = (select count(*) from photos);
```

Select user_id, username, count(*) as total_likes_per_user from users inner join likes on users.id = likes.user_id Group by likes.user_id having total_likes_per_hour = (select count(*) from photos);

INVESTOR METRICS

2) Identifying Bots and Fake Accounts

Investors want to know if the platform is crowded with fake and dummy accounts. **Task:** Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

Result/Output:

So, these **13 user accounts** with **user_id** and **total_likes_per_hours** are provided. These are the accounts which seems likes **Bots/Fake accounts.**

user_id	username	total_likes_per_user
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

Conclusion

- All the tasks which are given as Instagram User Analytics has been solved with appropriate results/output.
- For completing these tasks, I have used MySQL workbench 8.0
- All the images of the results are also provided.

Thank You