CAPSTONE PROJECT NO.2

INSTAGRAM USER ANALYTICS

By C.Manoj Kumaar

PROJECT DESCRIPTION

The objectives of this project are

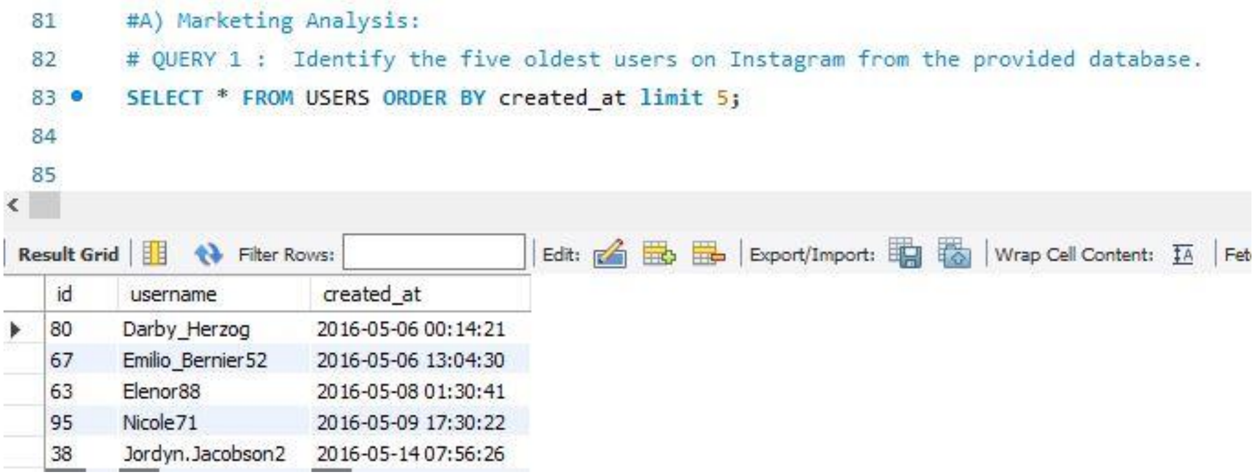
- To analyse the given dataset using MySQL workbench
- To conduct marketing analysis for rewarding the loyal user, engage inactive users, reward the most active user, identify most frequently used hashtags, and when to launch Ad compaigns in Instagram social media.
- To conduct Investor metrics for identifying the list of Bots and Fake accounts and degree of user engagement

APPROACH

The approach for all seven different queries are explained below

Query 1: Identify the five oldest users on Instagram from the provided database.

1. The user table is created with 3 column names – ID, username, created_at.
2. The table is sorted in descending order based on date provided in created_ at column using ORDER BY command.
3. The top 5 is listed out using LIMIT command
4. The SQL queries and output is shown in Fig.1



```
#A) Marketing Analysis:
# QUERY 1 :  Identify the five oldest users on Instagram from the provided database.
SELECT * FROM USERS ORDER BY created_at limit 5;


```

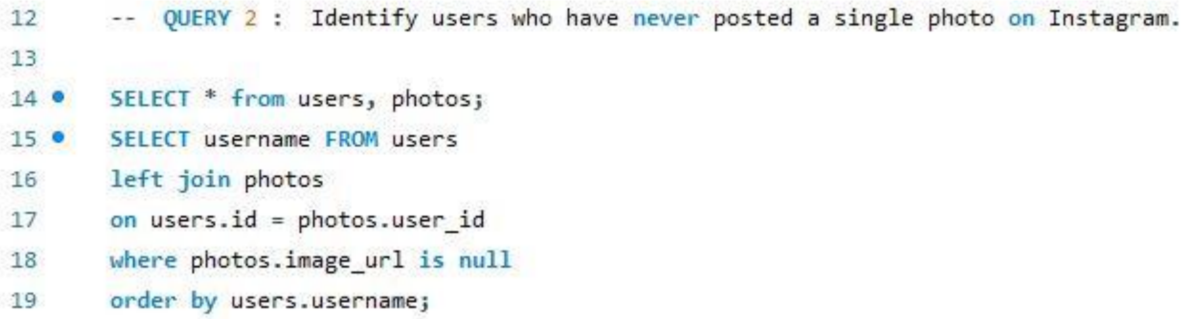
id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26

Fig 1. Query 1 – SQL commands and Output

Query 2 : Identify users who have never posted a single photo on Instagram.

1. Join ‘users’ and ‘photos’ table using LEFT JOIN method with reference to user id.

2. Filter the list using WHERE condition for image_url having NULL entries and ORDER BY command for username in ascending order as shown in Fig. 2 and the result is shown in Table 2.



```
--  QUERY 2 :  Identify users who have never posted a single photo on Instagram.

SELECT * from users, photos;
SELECT username FROM users
left join photos
on users.id = photos.user_id
where photos.image_url is null
order by users.username;
```

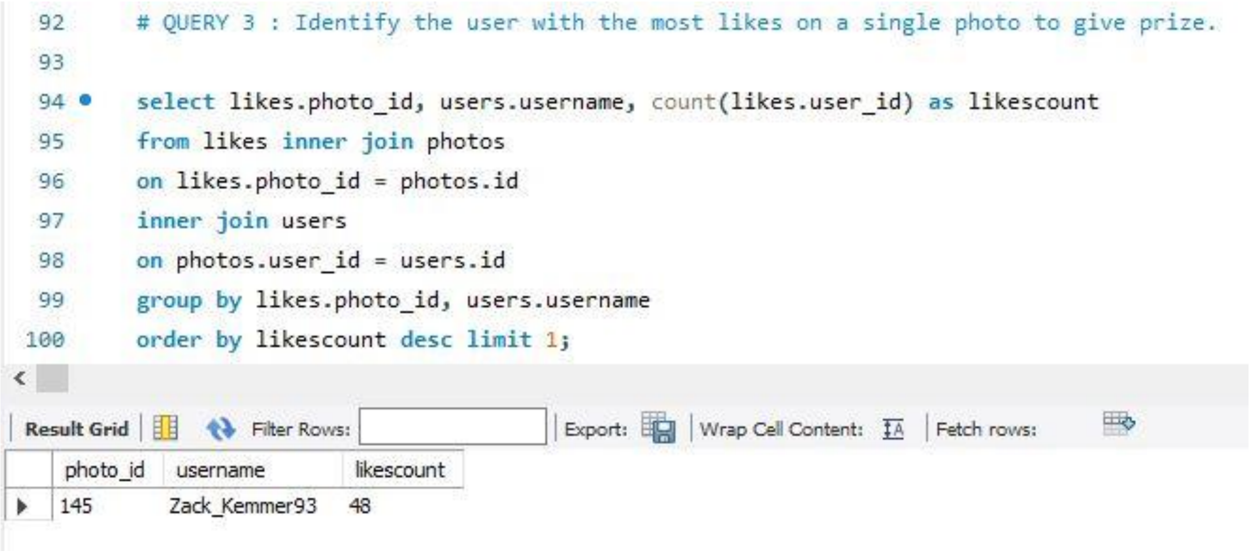
Fig 2 : SQL commands for query no.2

Table 1 : List of Inactive Users

Sl.No	username
1	Aniya_Hackett
2	Bartholome.Bernhard
3	Bethany20
4	Darby_Herzog
5	David.Osinski47
6	Duane60
7	Esmeralda.Mraz57
8	Esther.Zulauf61
9	Franco_Keebler64
10	Hulda.Macejkovic
11	Jaclyn81
12	Janelle.Nikolaus81
13	Jessyca_West
14	Julien_Schmidt
15	Kasandra_Homenick
16	Leslie67
17	Linnea59
18	Maxwell.Halvorson
19	Mckenna17
20	Mike.Auer39
21	Morgan.Kassulke
22	Nia_Haag
23	Ollie_Ledner37
24	Pearl7
25	Rocio33
26	Tierra.Trantow

QUERY 3 : Determine the winner of the contest and provide their details to the team.

1. All 3 tables - ‘likes’ , ‘photos’, and ‘users’ tables are joined using LEFT JOIN with reference to user id
2. User_id’s are counted for each photo_id (see line.99) and stored as ‘likescount’ (see lineno.94)
3. Photo_id of all users are listed in descending order and limited to 1 to find the user who got max number of likes for his photo as shown in Fig.3. He can be awarded with prize.



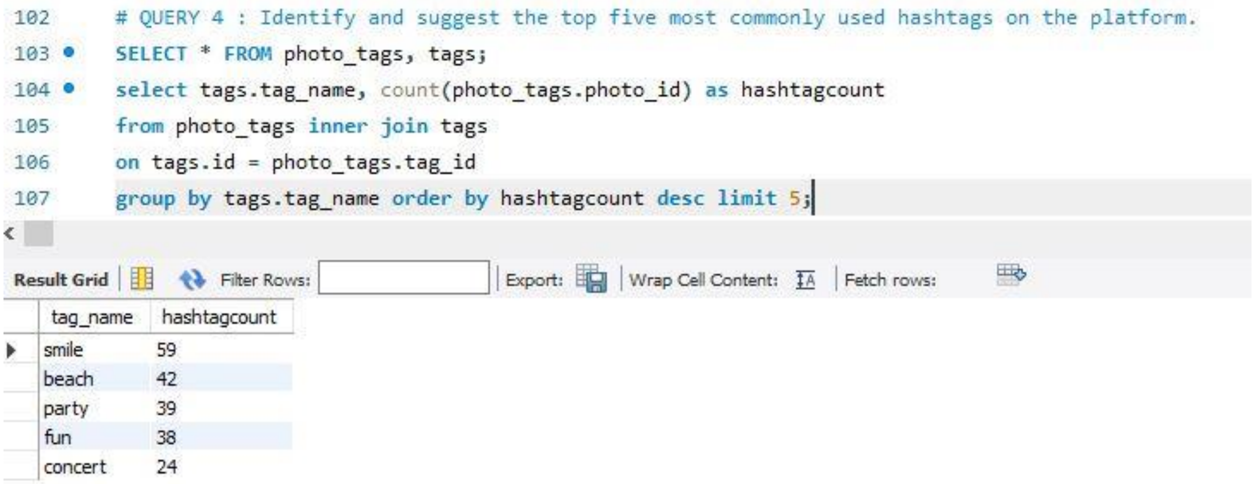
```sql
92      # QUERY 3 : Identify the user with the most likes on a single photo to give prize.
93
94      select likes.photo_id, users.username, count(likes.user_id) as likescount
95      from likes inner join photos
96      on likes.photo_id = photos.id
97      inner join users
98      on photos.user_id = users.id
99      group by likes.photo_id, users.username
100     order by likescount desc limit 1;
```

```
photo_id  username      likescount
145       Zack_Kemmer93 48
```

Fig 3: SQL commands for query 3.

QUERY 4: Identify and suggest the top five most commonly used hashtags on the platform.

1. Both ‘photo_tags’ and ‘tags’ tables are joined using INNER join method with reference to tag_id of each table.
2. Count the photo_id from photo_tags table as store it as hashtagcount according to tag_name.
3. List the first 5 tag names using LIMIT command.
4. Refer the queries and result in fig.4



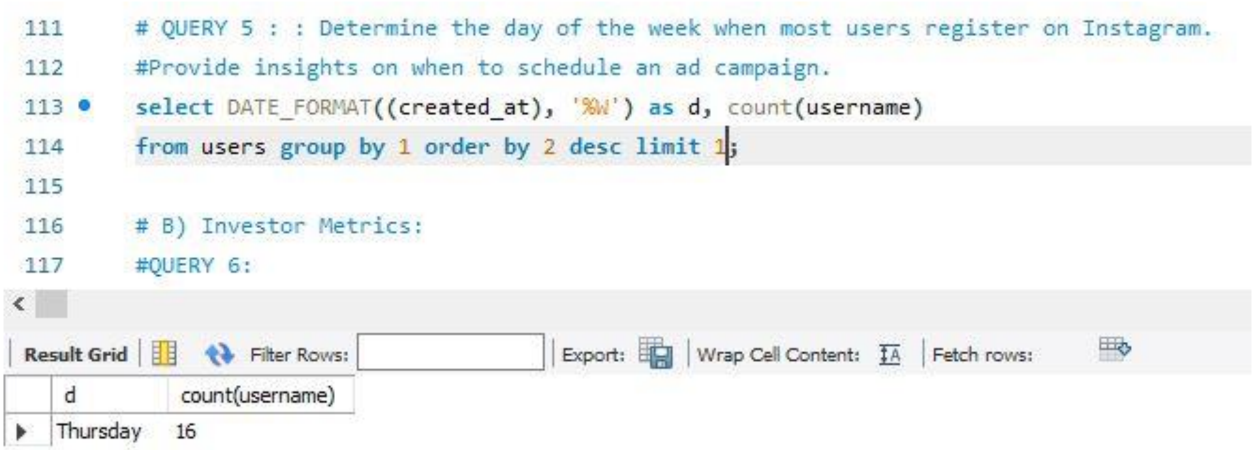
```sql
# QUERY 4 : Identify and suggest the top five most commonly used hashtags on the platform.
SELECT * FROM photo_tags, tags;
select tags.tag_name, count(photo_tags.photo_id) as hashtagcount
from photo_tags inner join tags
on tags.id = photo_tags.tag_id
group by tags.tag_name order by hashtagcount desc limit 5;
```

```
tag_name  hashtagcount
smile     59
beach     42
party     39
fun       38
concert   24
```

Fig 4: SQL commands for query 4

QUERY 5: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign..

1. Convert the date format of created_at column of table ‘users’ by ‘%W’ function as store it in variable ‘d’. Also, count the usernames joined in each day
2. List the days according to each day of week in descending order of username count and limit to one record.
3. Refer the queries and result in fig.5



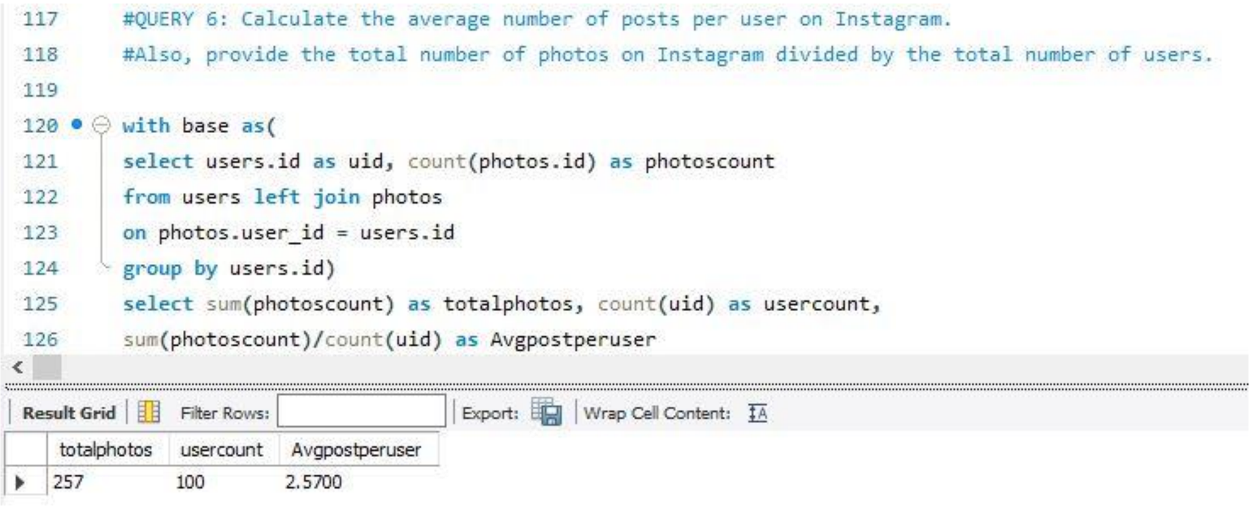
```
# QUERY 5 : : Determine the day of the week when most users register on Instagram.
#Provide insights on when to schedule an ad campaign.
select DATE_FORMAT((created_at), '%W') as d, count(username)
from users group by 1 order by 2 desc limit 1;

# B) Investor Metrics:
#QUERY 6:
```

d	count(username)
Thursday	16

Fig 5: SQL commands for query 5

Query 6: 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.

1. Both ‘users’ and ‘photos’ tables are joined using LEFT JOIN with reference to user id
2. A temporary structure is created within base function in which total number of photos and total number of users are determined using count function.
3. Average post per user is calculated by dividing total no. of photos by total no. of users
4. Refer the queries and result in fig.6



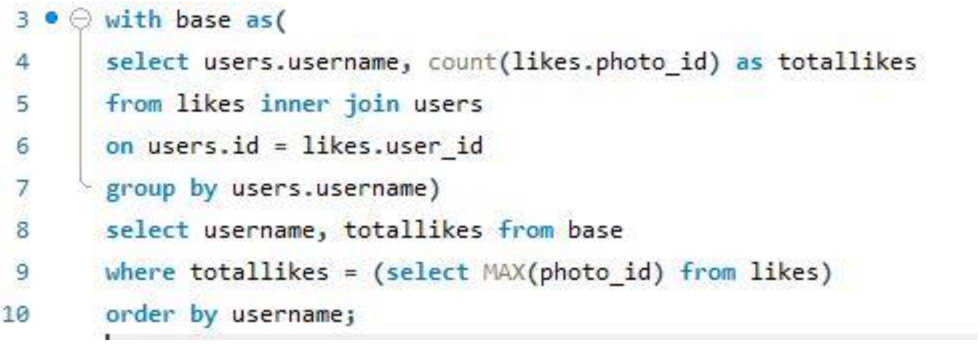
```sql
#QUERY 6: Calculate the average number of posts per user on Instagram.
#Also, provide the total number of photos on Instagram divided by the total number of users.

with base as(
select users.id as uid, count(photos.id) as photoscount
from users left join photos
on photos.user_id = users.id
group by users.id)
select sum(photoscount) as totalphotos, count(uid) as usercount,
sum(photoscount)/count(uid) as Avgpostperuser
```

```
totalphotos   usercount   Avgpostperuser
257           100         2.5700
```

Fig 6: SQL commands for query 6

Query 7 : Identify the list of Bots or Fake accounts who likes all the posts in Instagram

5. Both ‘users’ and ‘likes’ tables are joined using INNER JOIN with reference to user id
6. A temporary structure is created within base function in which total number of likes given by each user using count function and group by function.
7. Usernames whose total likes are equal to total no. of photos / posts are listed as shown in fig 7 and Table 2.



```sql
with base as(
select users.username, count(likes.photo_id) as totallikes
from likes inner join users
on users.id = likes.user_id
group by users.username)
select username, totallikes from base
where totallikes = (select MAX(photo_id) from likes)
order by username;
```

Fig. 7 : SQL commands for Query 7

Table 2 : List of Bots/ fake accounts

Sl. No.	username	total likes
1	Aniya_Hackett	257
2	Bethany20	257
3	Duane60	257
4	Jaclyn81	257

5	Janelle.Nikolaus81	257
6	Julien_Schmidt	257
7	Leslie67	257
8	Maxwell.Halvorson	257
9	Mckenna17	257
10	Mike.Auer39	257
11	Nia_Haag	257
12	Ollie_Ledner37	257
13	Rocio33	257

TECH-STACK USED :

MySQL workbench is used for the following reasons

- All tables can be stored based on logical grouping of data
- Access and manipulate data quickly and effectively
- Make changes in a single location that can reflect in multiple tables.

INSIGHTS

1. 13% of users are fake in instagram. They can be removed from instagram when they fail to authenticate with their selfie photo in their accounts. So that duplicate accounts with same person can deleted.
2. 25% of users are inactive in instagram. They can be made active by sending them promotional mails.
3. 63% of users are active in instagram. To keep them active, regular contests can be conducted.
4. Most of the users are joined on Thursdays and ad compaigns can be conducted on Thursdays
5. Posts can be outreached more by using the frequently used hash_tags identified.

RESULT

This project helped me to conduct marketing analysis and investor metrics based on SQL queries framed in MySQL. The dataset was clean and clear to understand and carry out the project with ease.