

Project no 2

INSTAGRAM USER ANALYTICS

SQL Fundamentals

Project description:-

The main purpose of the Instagram user analytics is to derive a conclusion as to how to improve the user interaction among its users and also to provide the companies the data required for them to use the platform for marketing.

Approach:-

The SQL queries were used to solve the required tasks of the project which were taught in the learning. Instead of converting the word file to flat text csv I have just copied and pasted the entire dataset into the workbench which created the database then started working with it.

Tech-Stack Used:-

I have used MYSQL80 the latest version of the MYSQL workbench as suggested in the project, following all the setups all the required installations have been done.

Analysis performed.

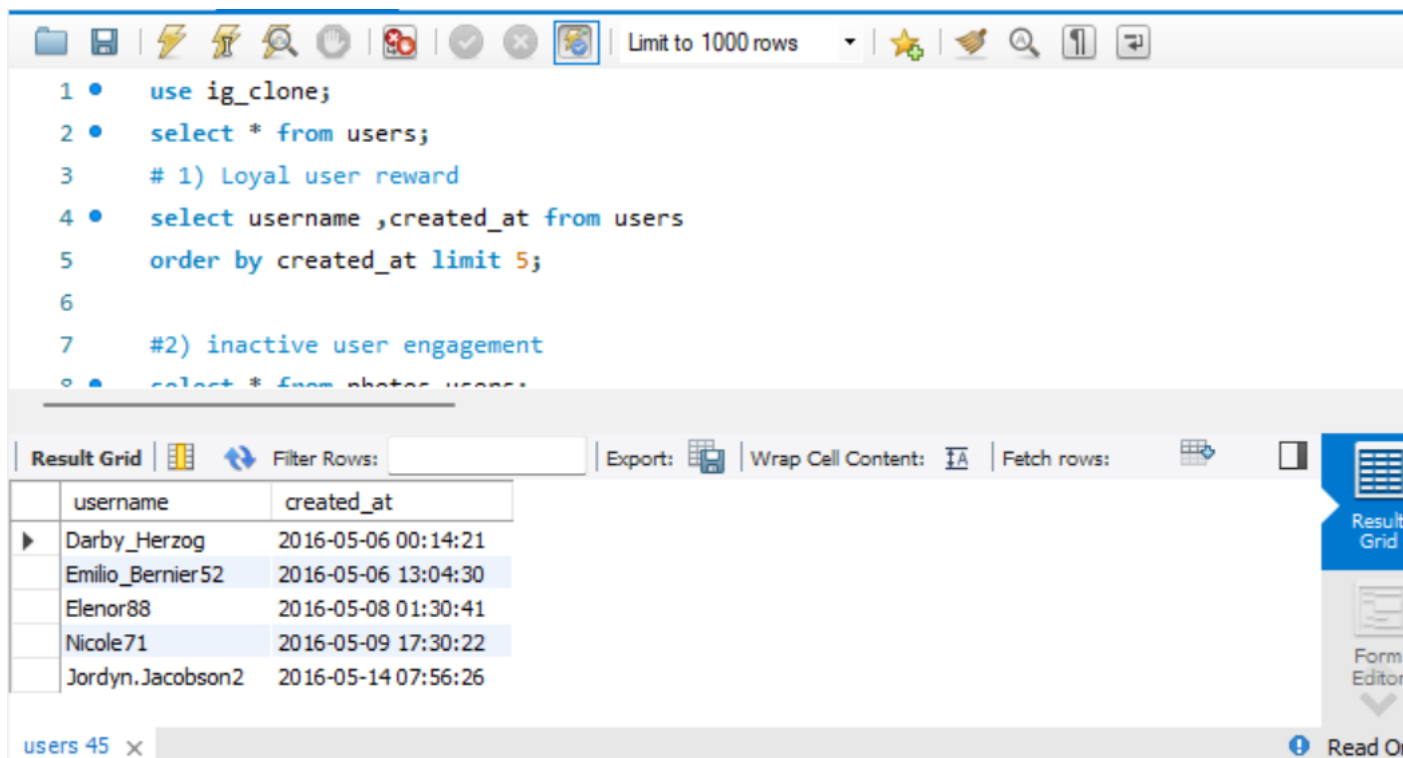
SQL Tasks

A) Marketing analysis;

1) Loyal User reward;

To get the oldest users the created_at from the users table is used and sorted in ascending order through registered dates.

The list is limited to 5 to get the five old users.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • use ig_clone;
2 • select * from users;
3   # 1) Loyal user reward
4 • select username ,created_at from users
5   order by created_at limit 5;
6
7   #2) inactive user engagement
8 • select * from photos users;
```

Below the editor is the 'Result Grid' section, which displays the query results in a table. The table has two columns: 'username' and 'created_at'. The results are as follows:

username	created_at
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

At the bottom of the interface, there is a tab labeled 'users 45' and a 'Read O' button.

2)Inactive user engagement

We have used the left join function from photos table to users table to get the users with null posts.

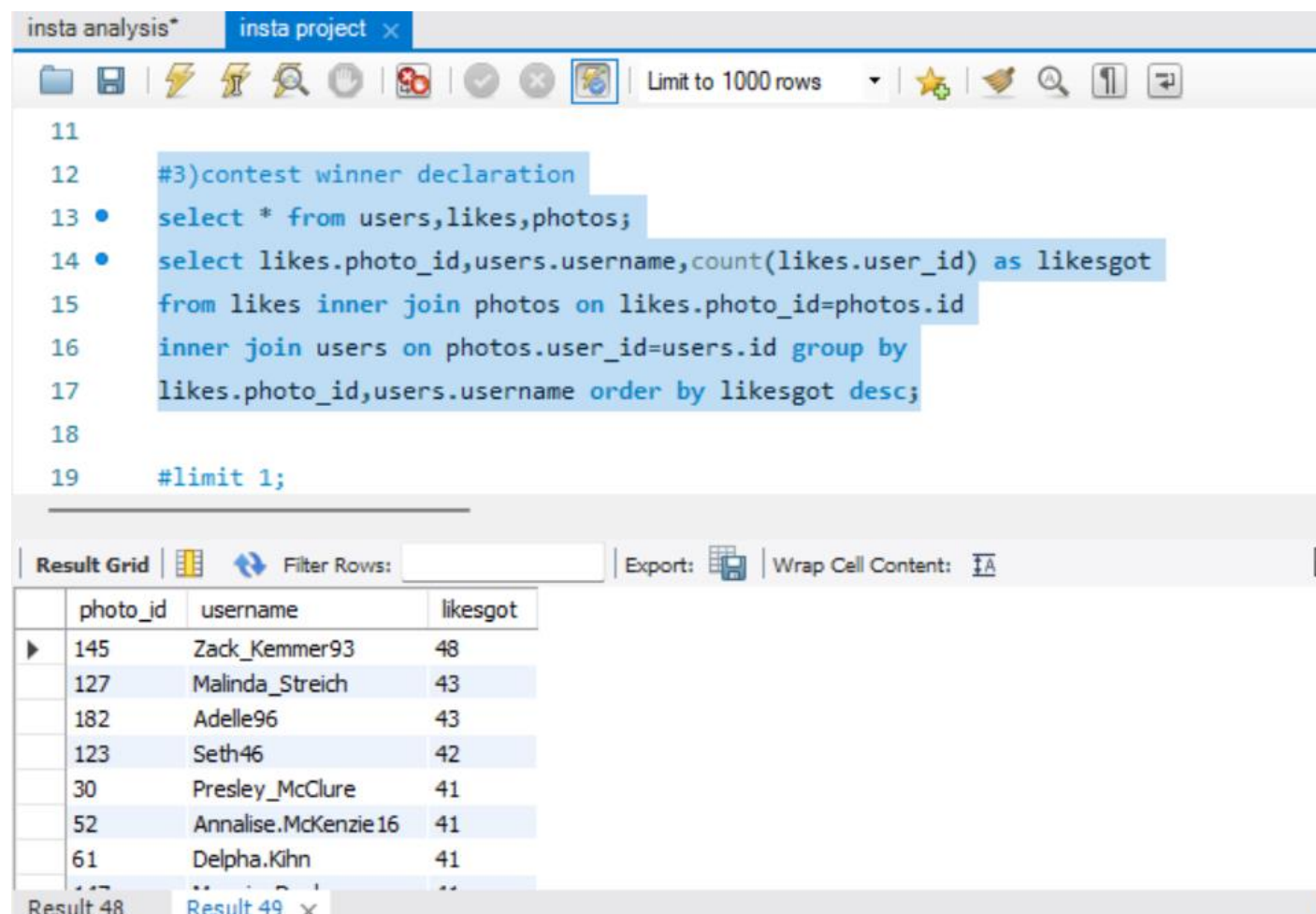
```
5      order by created_at limit 5;
6
7      #2) inactive user engagement
8 •    select * from photos,users;
9 •    select * from users u left join photos p on p.user_id=u.id
10     where p.image_url is null order by username;
11
12     #3)contest winner declaration
13 •    select * from users,likes,photos;
```

	A	B	C	D	E	F	G	H
	id	username	created_at	id	image_url	user_id	created_at	
1	5	Aniya_Har	#####	NULL	NULL	NULL	NULL	
2	83	Bartholon	#####	NULL	NULL	NULL	NULL	
3	91	Bethany20	#####	NULL	NULL	NULL	NULL	
4	80	Darby_He	#####	NULL	NULL	NULL	NULL	
5	45	David.Osi	#####	NULL	NULL	NULL	NULL	
6	54	Duane60	#####	NULL	NULL	NULL	NULL	
7	90	Esmerald	#####	NULL	NULL	NULL	NULL	
8	81	Esther.Zul	#####	NULL	NULL	NULL	NULL	
9	68	Franco_K	#####	NULL	NULL	NULL	NULL	
0	74	Hulda.Ma	#####	NULL	NULL	NULL	NULL	
1	14	Jaclyn81	#####	NULL	NULL	NULL	NULL	
2	76	Janelle.Ni	#####	NULL	NULL	NULL	NULL	
3	89	Jessyca_V	#####	NULL	NULL	NULL	NULL	
4	57	Julien_Sch	#####	NULL	NULL	NULL	NULL	
5	7	Kassandra	#####	NULL	NULL	NULL	NULL	
6	75	Leslie67	#####	NULL	NULL	NULL	NULL	
7	53	Linnea59	#####	NULL	NULL	NULL	NULL	
8	24	Maxwell.F	#####	NULL	NULL	NULL	NULL	
9	41	Mckenna1	#####	NULL	NULL	NULL	NULL	
0	66	Mike.Auer	#####	NULL	NULL	NULL	NULL	
1	49	Morgan.K	#####	NULL	NULL	NULL	NULL	
2	71	Nia_Haag	#####	NULL	NULL	NULL	NULL	
3	36	Ollie_Ledr	#####	NULL	NULL	NULL	NULL	
4	34	Pearl7	#####	NULL	NULL	NULL	NULL	
5	21	Rocio33	#####	NULL	NULL	NULL	NULL	
6	25	Tierra.Tra	#####	NULL	NULL	NULL	NULL	
7								
8								
9								

Here are the list of users with zero photos uploaded.

3)Contest winner declaration:

All the 3 tables i.e users,likes,photos are taken and the number of likes for a picture is counted using the count function and inner join through the photo_id which is unique for each photo uploaded.



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
11
12 #3)contest winner declaration
13 • select * from users,likes,photos;
14 • select likes.photo_id,users.username,count(likes.user_id) as likesgot
15 from likes inner join photos on likes.photo_id=photos.id
16 inner join users on photos.user_id=users.id group by
17 likes.photo_id,users.username order by likesgot desc;
18
19 #limit 1;
```

The results grid displays the following data:

	photo_id	username	likesgot
▶	145	Zack_Kemmer93	48
	127	Malinda_Streich	43
	182	Adelle96	43
	123	Seth46	42
	30	Presley_McClure	41
	52	Annalise.McKenzie16	41
	61	Delpha.Kihn	41
	117	Malinda_Streich	41

Here zack_kemmer93 had the highest number of likes for a photo which makes him the clear winner of the contest.

4)Hashtag research

The count function is used to count the maximum used hashtags from each photo uploaded the hashtags are collected and then check the hashtag name in the hashtag table.

The screenshot shows a data analysis tool interface with a tab labeled "insta project". The main area contains a SQL query:

```
20
21
22 #4)hashtag research
23 • select * from tags,photo_tags;
24 • select tags.tag_name,count(photo_tags.photo_id) as hashtag from photo_tags inner join
25 tags on tags.id=photo_tags.tag_id group by tags.tag_name order by hashtag desc;
26 #limit 5;
27
28 #5)ad campaign launch
```

Below the query, the "Result Grid" is displayed, showing a table with two columns: "tag_name" and "hashtag". The results are as follows:

tag_name	hashtag
smile	59
beach	42
party	39
fun	38
concert	24
food	24
lol	24

The interface also includes a "Filter Rows" section, an "Export" button, and a "Wrap Cell Content" option. On the right side, there are buttons for "Result Grid", "Form Editor", and "Read Only".

5)Ad campaign launch

The day which is most used by the users to enter the platform is calculated using the created_at column from the users table and also the date_format is used to derive the specific day of the date here.the list is ordered from highest joins to lowest.

```
26 #limit 5;
27
28 #5)ad campaign launch
29 • select * from users;
30 • select DATE_FORMAT(created_at,'%W') as dayy ,count(username)
31    from users group by 1 order by 2 desc;
32
33
34 #Inverstor metrics
```

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

	dayy	count(username)
▶	Thursday	16
	Sunday	16
	Friday	15
	Tuesday	14
	Monday	14
	Wednesday	13
	Saturday	12

users 52 | Result 53 x | Read Only

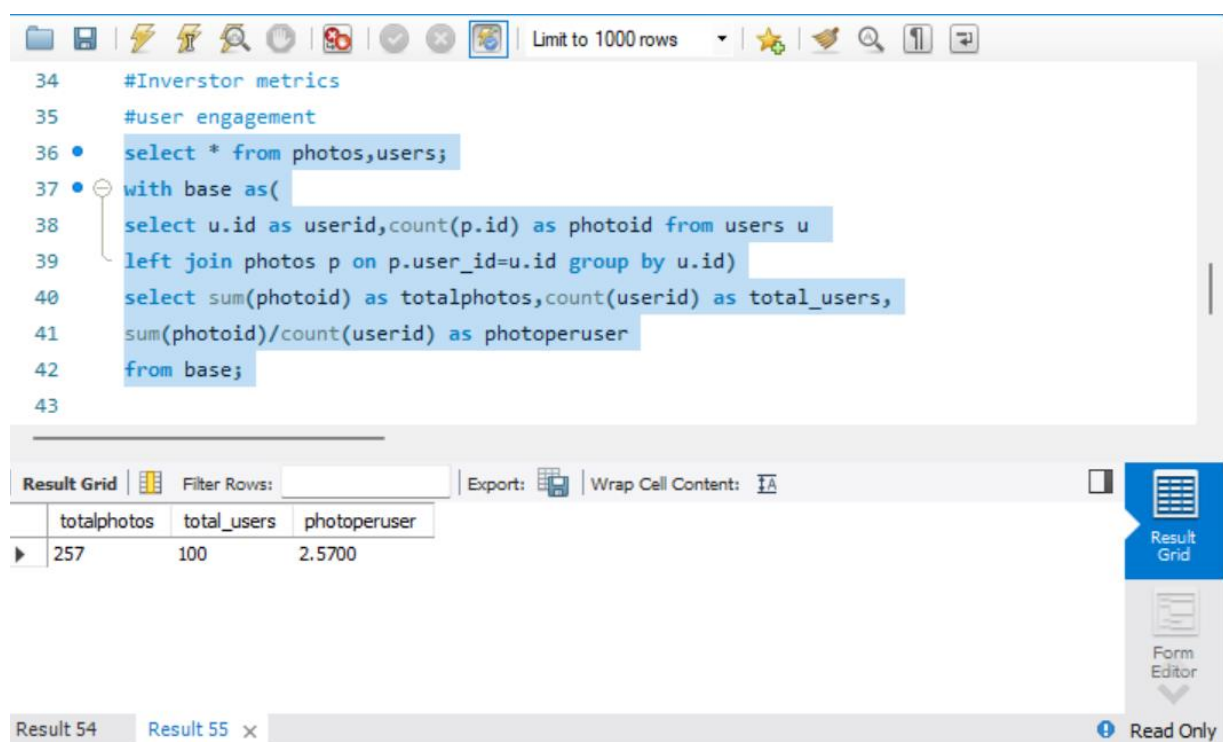
B)Investor metrics

1)User engagement:

Here the average number of posts has to be calculated from the users table the total number of users is 100 while total photos posted are 257 which gives an average of 2.57.

This is derived using the SQL queries

Total photos are counted using sum function and left joined to users in the user table and then divided to get average



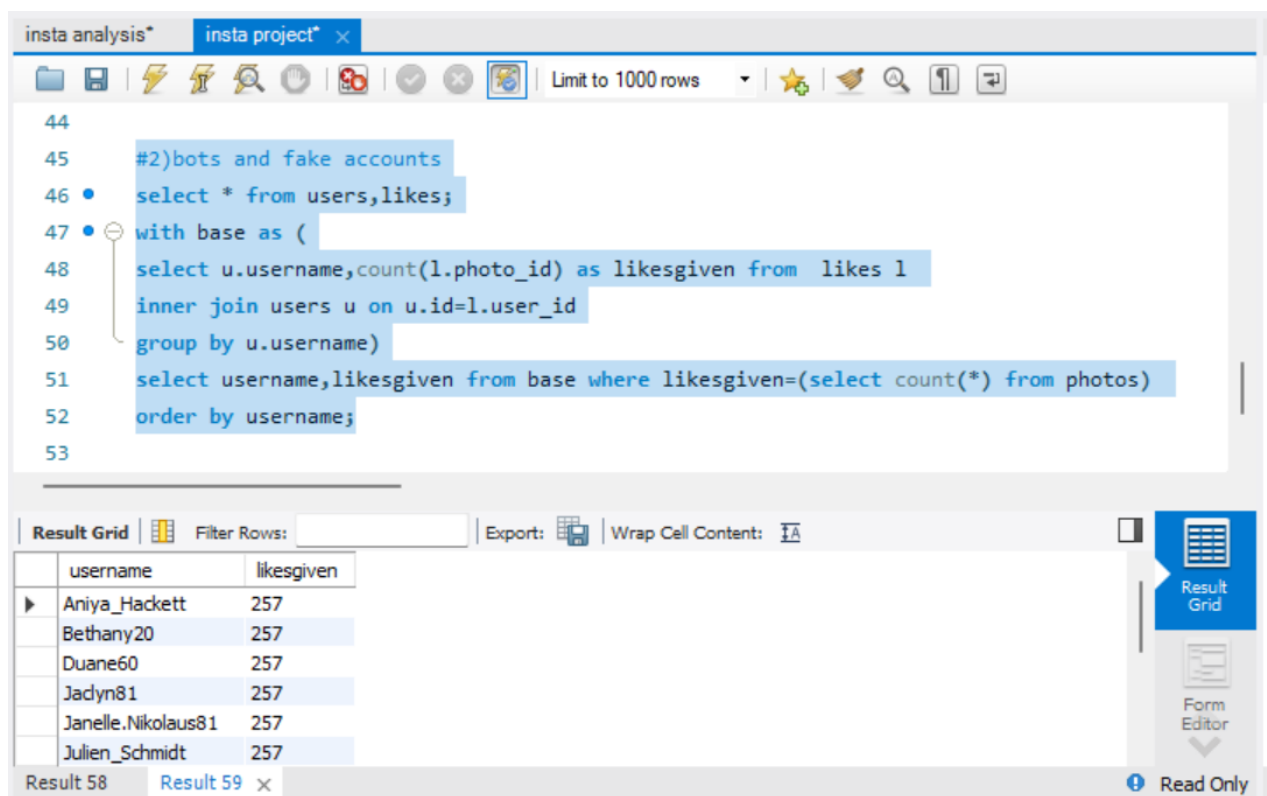
```
34 #Investor metrics
35 #user engagement
36 • select * from photos,users;
37 • with base as(
38   select u.id as userid,count(p.id) as photoid from users u
39   left join photos p on p.user_id=u.id group by u.id)
40   select sum(photoid) as totalphotos,count(userid) as total_users,
41   sum(photoid)/count(userid) as photoperuser
42   from base;
43
```

totalphotos	total_users	photoperuser
257	100	2.5700

Result 54 Result 55 x Read Only

2)Bots are fake accounts:

From the users table and likes table the users who have liked all the photos posted are calculated since the total photos posted summed up to be 257 the users who have liked all 257 photos are listed and grouped by the usernames. It turned out to be there are 13 bots or fake accounts



```
44
45 #2)bots and fake accounts
46 • select * from users,likes;
47 • with base as (
48   select u.username,count(l.photo_id) as likesgiven from likes l
49   inner join users u on u.id=l.user_id
50   group by u.username)
51   select username,likesgiven from base where likesgiven=(select count(*) from photos)
52   order by username;
53
```

username	likesgiven
Aniya_Hackett	257
Bethany20	257
Duane60	257
Jadyn81	257
Janelle.Nikolaus81	257
Julien_Schmidt	257

Result 58 Result 59 x

Read Only

Insights:

The process of left join, inner join and then counting the total number of summing up gave me the skills to deal with the relational databases.

The dateformat function which converted the day number of week into day of the week was helpful

Summing and counting functions helped in completing the tasks.

Result:

Completed all the tasks using SQL. This task provided me with knowledge of MySQL queries and how to deal with the relational databases