

Instagram user Analytics Project-2

(Trainity Assignment)

SQL Task:

A) Marketing Analysis:

1. **Loyal user reward:** Identify the five oldest users on Instagram from the provided data.

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'ig_clone' database selected. The 'Query' editor contains the following SQL query:

```
1 SELECT * from ig_clone.users
2 ORDER BY created_at ASC
3 LIMIT 5;
```

The 'Result Grid' shows the output of the query:

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

Answer: In this task we first use the ig_clone database 's table users to identify the five oldest users using select query. and arrange the data according to the created_at (date of account creation) in ascending order. And limit the data up to 5 because we need the 5 oldest users.

2. **Inactive User Engagement:** Identify users who have never posted a single photo on Instagram.

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'ig_clone' database selected. The 'Query' editor contains the following SQL query:

```
1 SELECT * FROM users,photos;
2 select * from users u left join photos p
3 on p.user_id = u.id
4 where p.image_url is null
5 Order by u.username;
```

The 'Result Grid' shows the output of the query:

id	username	created_at	id	image_url	user_id	created_at
5	Ariva_Hackett	2016-12-07 01:04:39				
83	Bartholome.Bernhard	2016-11-06 02:31:23				
91	Bethany20	2016-06-03 23:31:53				
80	Darby_Herzog	2016-05-06 00:14:21				
45	David.Osinski47	2017-02-05 21:23:37				
54	Duane60	2016-12-21 04:43:38				
90	Esmeralda.Mraz57	2017-03-03 11:52:27				
81	Esther.Zulauf61	2017-01-14 17:02:34				
68	Franco_Keebler64	2016-11-13 20:09:27				
74	Hulda.Macejkovic	2017-01-25 17:17:38				
14	Jaclyn81	2017-02-06 23:29:16				
76	Janelle.Nikolaus81	2016-07-21 09:26:09				
89	Jessica_West	2016-09-14 23:47:05				
57	Julien_Schmidt	2017-02-02 23:12:48				
7	Kassandra_Homenick	2016-12-12 06:50:08				
75	Ledies7	2016-09-21 05:14:01				
53	Limes59	2017-02-07 07:49:24				
24	Maxwell.Holvorson	2017-04-18 02:32:44				
41	McKenzie17	2016-07-17 17:25:45				
66	Mike.Auer39	2016-07-01 17:36:15				
49	Morgan.Kassulke	2016-10-30 12:42:31				
71	Nia_Haag	2016-05-14 15:36:50				
36	Oliver.Ledner37	2016-08-04 15:42:20				

Answer: In this, task we will need the users table and photos table from database lg_clone.

Using the select query we join the user (u) and Photos (p) and photo's user_id will be equal to user's id. Using WHERE clause we will set Image_url as null (to find who have zero post) and arrange in the order by username so we will get users name who have posted zero post.

3. **Contest Winner Declaration:** Determine the winner of the contest and provide their details to the team. (find the most liked single photo).

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 select * from Likes,photos;
2 select likes.photo_id,users.username, count(likes.user_id) as nooflikes
3 from likes
4 inner join photos on likes.photo_id = photos.id
5 inner join users on photos.user_id = users.id
6 group by likes.photo_id,users.username
7 order by nooflikes desc ;
```

The Result Grid shows the following data:

photo_id	username	nooflikes
145	Zack_Kammer93	48
127	Malinda_Streich	43
182	Adelle96	43
123	Seth46	42
30	Presley_McClune	41
52	Annaliese_McKenzie16	41
61	DelphaJGhm	41
147	Meagie Doyle	41

Answer: In this, task we will need likes, users and photos table from database.

Using Select query we will fetch Photo_id from likes & username from user's table.

using Count clause we will count the user_id as no. of likes from likes table.

now, we will inner join likes, photos and users. where from Likes's photo_id is equal to photo's id and photo's user_id equal to user's id. Now we will group them by photo_id and username and arrange them in order by no of likes in descending. To get the most liked single photo.

4. **Hashtag Research:** Identify and suggest the top five most commonly used hashtags on the platform.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 select * from photo_tags, tags;
2 select t.tag_name, count(p.photo_id) as ht
3 from photo_tags p inner join tags t on t.id = p.tag_id
4 group by t.tag_name order by ht desc
5 limit 5;
```

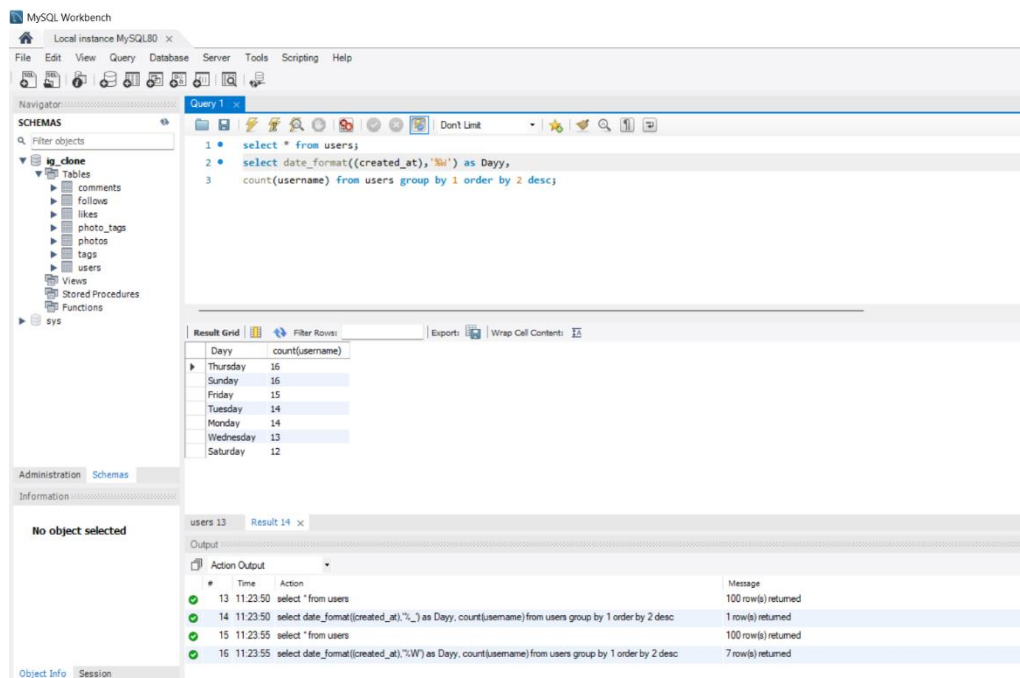
The Result Grid shows the following data:

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

Answer: In this, task we will need photo_tags and tags table from the database. using the SELECT query we will get the tag_name from **Tag** table and count the photo_id from **Photo_tags** table as (ht) **Hastag**. From the **photo_tag** (p is a short name for photo_tag) and we will inner join **tags** (t is the short name for tags).

The id of (t) is equal to the tag_id of (p) and group them by tag_name from (t) and arrange them in order by (ht) **hastag** in descending. Limit it up to 5 because we need only five tags.

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



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', and 'users'. The main query editor contains the following SQL code:

```
1 select * from users;
2 select date_format(created_at, '%W') as Dayy,
3 count(username) from users group by 1 order by 2 desc;
```

The 'Result Grid' shows the output of the second query, displaying the day of the week and the count of users:

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

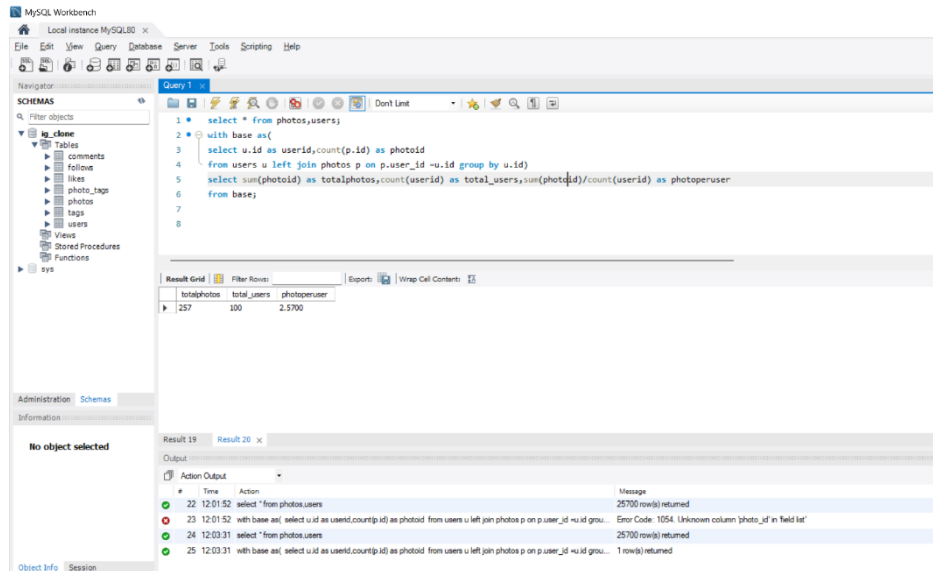
The bottom panel shows the 'Output' tab with a log of actions and their results:

#	Time	Action	Message
13	11:23:50	select * from users	100 row(s) returned
14	11:23:50	select date_format(created_at, '%W') as Dayy, count(username) from users group by 1 order by 2 desc	1 row(s) returned
15	11:23:55	select * from users	100 row(s) returned
16	11:23:55	select date_format(created_at, '%W') as Dayy, count(username) from users group by 1 order by 2 desc	7 row(s) returned

Answer: In this, task we will need **USERS** table from the database. Using the Date_format (...,'%W') we Will convert the date of creation into day like (Monday, Tuesday...). And we will count the Usernames from the user table. Group all the data by the first column as **DAY** and arranging All the data order by username in Descending.

B) Investor Metrics:

1. **User Engagement:** 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.



The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'ig_clone' database with tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', 'users', 'Views', 'Stored Procedures', 'Functions', and 'sys'. The 'Query' editor contains the following SQL query:

```
1 select * from photos,users;
2 with base as(
3   select u.id as userid,count(p.id) as photoid
4   from users u left join photos p on p.user_id =u.id group by u.id)
5 select sum(photoid) as totalphotos,count(userid) as total_users,sum(photoid)/count(userid) as photoperuser
6 from base;
```

The 'Result Grid' shows the following data:

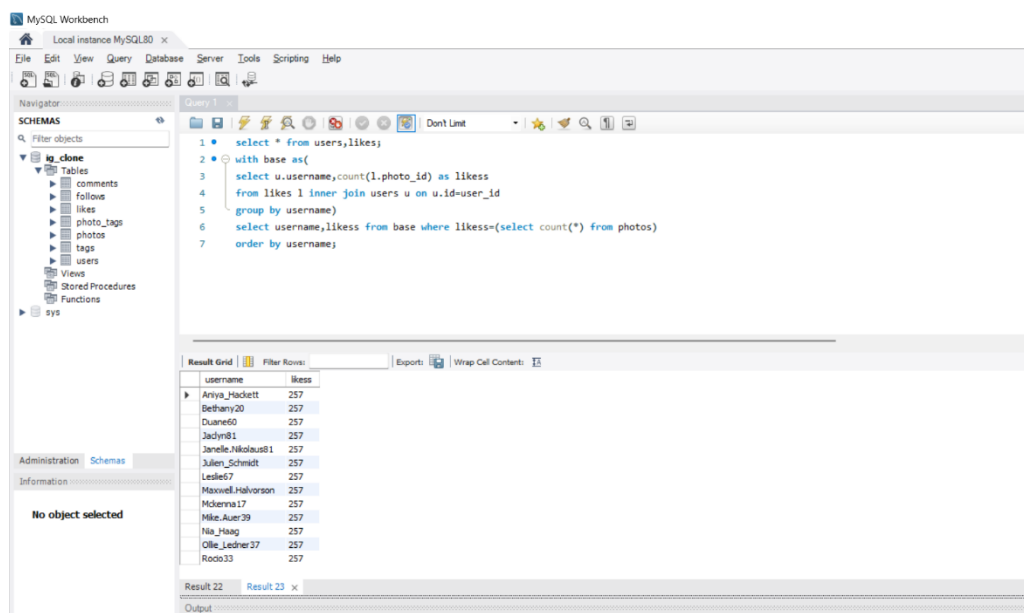
totalphotos	total_users	photoperuser
257	100	2.5700

The 'Output' pane shows the execution log with the following messages:

- 22 12:01:52 select * from photos,users 25700 row(s) returned
- 23 12:01:52 with base as(select u.id as userid,count(p.id) as photoid from users u left join photos p on p.user_id =u.id group by u.id) Error Code: 1054. Unknown column 'photo_id' in 'field list'
- 24 12:03:31 select * from photos,users 25700 row(s) returned
- 25 12:03:31 with base as(select u.id as userid,count(p.id) as photoid from users u left join photos p on p.user_id =u.id group by u.id) 1 row(s) returned

Answer: In this, task we will need photos, users table from database. Then we will create an temporary table named as base. using select query **u.id** (user table's id column) renamed as **userid** and we will count the **p.id** (photo's id) and renamed as **photoid** from using **Left Join** we will join each user with there photos which will lead you to get every single users photos even with 0 photos and group them by user id. then we will sum all the **photo id** and named as total photos and count the user id and named as total users then we will divide the total photos with the counted users and named as photo per user from the base.

2. **Bots & Fake Accounts:** Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.



The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'ig_clone' database with tables like 'comments', 'follows', 'likes', 'photo_tags', 'photos', 'tags', 'users', 'Views', 'Stored Procedures', 'Functions', and 'sys'. The 'Query' editor contains the following SQL query:

```
1 select * from users,likes;
2 with base as(
3   select u.username,count(1.photo_id) as likess
4   from likes l inner join users u on u.id=users_id
5   group by username)
6 select username,likess from base where likess=(select count(*) from photos)
7 order by username;
```

The 'Result Grid' shows the following data:

username	likess
Aniya_Hackett	257
Bethany20	257
Duane60	257
Jadyn81	257
Janelle.Nikolaus81	257
Julien_Schmidt	257
Leslie7	257
Maxwell.Halvorson	257
McKenzie17	257
Mike_Auer39	257
Nia_Hoag	257
Offie.Ledner27	257
Rocio33	257

The 'Output' pane shows the execution log with the following messages:

- 22 12:01:52 select * from users,likes 25700 row(s) returned
- 23 12:01:52 with base as(select u.username,count(1.photo_id) as likess from likes l inner join users u on u.id=users_id group by username) Error Code: 1054. Unknown column 'users_id' in 'field list'
- 24 12:03:31 select username,likess from base where likess=(select count(*) from photos) 25700 row(s) returned
- 25 12:03:31 order by username 1 row(s) returned

Answer: In this, task we will use user and likes table from database. With creating an temporary table called **base**, then we will get the username from **users** table and count the photo_id from table **likes** and name it as likess.

Later we will join the the id from user table and user_id from photos table. We will get the each likes done by a single user and group them by username and there likes.