



# Instagram User Analysis

# Project description

The purpose of this project is analysing user interactions and engagement with the Instagram app in order to offer insightful information that can support the expansion of the company.

Monitoring how consumers interact with digital products, like software or mobile apps, is known as user analysis. Several teams within the company can make use of the insights obtained from this investigation. For instance, the product team may use these insights to choose which new features to develop, the marketing team may use them to start a new campaign, and the development team may use them to enhance the user experience in general.

In order to evaluate Instagram user data and respond to inquiries from the management team, we will be utilizing SQL and MySQL Workbench for this project. The product manager and the other team members will use your insights to make well-informed decisions regarding the Instagram app's future course.



# Approach

Firstly I installed MYSQL software and with the help of the videos provided by trainity I was able to build my basics on the software and relational database MYSQL.

After installing I created the database "ig\_clone" using the command provided, and visually analysed each and every command's functionality. What tables are being created, what are the attributes and inputs used, how can one data be linked to another.

## Tech used

For this project, I have installed a Windows 11 vm on my mac using UTM where in I installed the software - MySQL Workbench 8.0.

I have been using the same for all the analysis as it is a versatile tool that offers several advantages for data analysis projects like-

1. User friendly interface
2. Powerful editor
3. Great visualisation
4. Compatible with large datasets
5. Secure

I have also used Microsoft Excel.

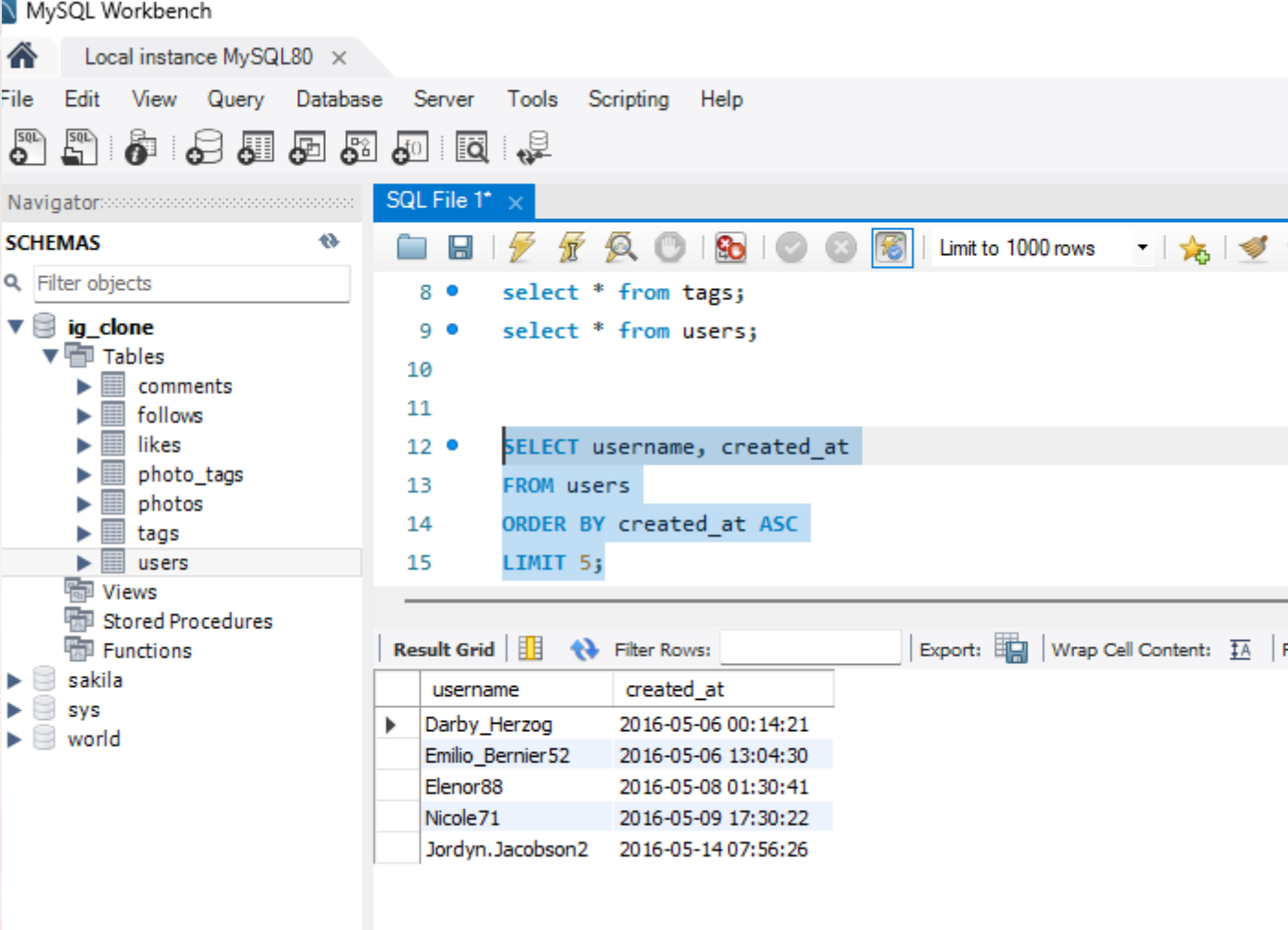
# Insights

## Knowledge and Understanding Acquired from Instagram User Analysis: User Engagement Trends

1. Users that regularly shared excellent visual content with popular hashtags showed high levels of interaction.
2. Relevant and trending hashtags have a big impact on post interaction and reach. The significance of hashtag strategy was shown by the fact that the majority of user interactions were attributed to the top five hashtags.
3. In terms of likes and in order to increase engagement, visual appeal and relatable descriptions were essential.
4. It was found that the number of followers increased in direct proportion to the frequency of posting.
5. The perceived worth of the prizes and the simplicity of the contest rules were associated with distinct patterns of participation.
6. MySQL Workbench offered a smooth environment for effective data analysis and querying. Performance was enhanced by using indexes and optimizing queries, particularly when examining huge datasets.
7. Trending Hashtags Are Important
8. Activity of the users and their contribution to the platform increases engagement.

# Result

**Task 1-** Identify the five oldest users on Instagram from the provided database.

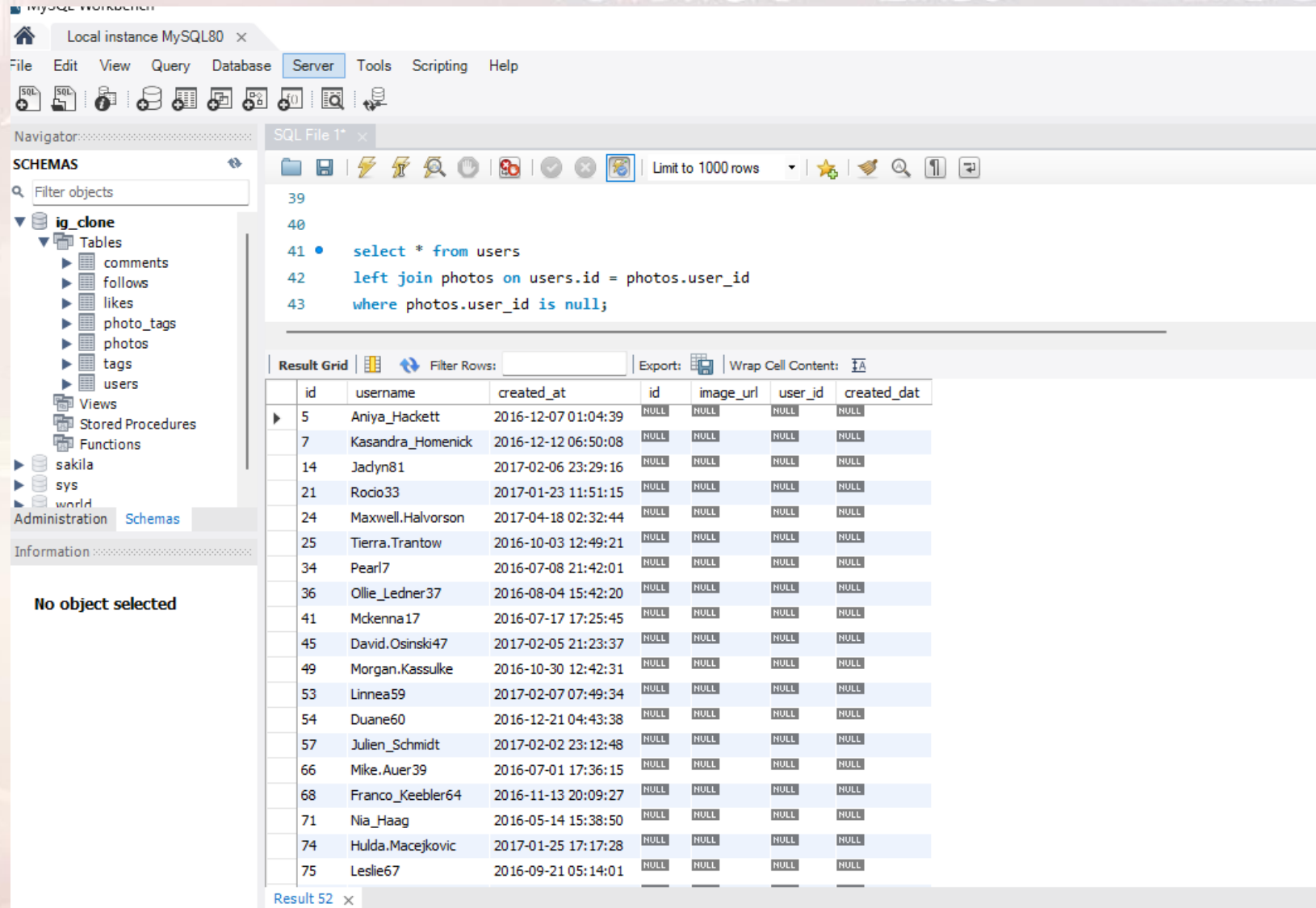


The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'ig\_clone' with several tables, including 'users'. The main editor window displays a SQL query: `SELECT username, created_at FROM users ORDER BY created_at ASC LIMIT 5;`. Below the query, the 'Result Grid' shows the top five oldest users based on their 'created\_at' timestamp.

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

# Result

## Task 2- Identify users who have never posted a single photo on Instagram.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'ig\_clone' selected, showing tables like 'comments', 'follows', 'likes', 'photo\_tags', 'photos', 'tags', and 'users'. The main area shows a SQL query in the 'SQL File 1\*' tab:

```
39
40
41 • select * from users
42 left join photos on users.id = photos.user_id
43 where photos.user_id is null;
```

The 'Result Grid' shows the results of the query, displaying a list of users who have never posted a photo. The columns are: id, username, created\_at, id, image\_url, user\_id, and created\_at. The results are as follows:

	id	username	created_at	id	image_url	user_id	created_at
▶	5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
	7	Kassandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
	14	Jadyn81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
	21	Rodo33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
	24	Maxwell.Halvorson	2017-04-18 02:32:44	NULL	NULL	NULL	NULL
	25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
	34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
	36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
	41	Mckenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL
	45	David.Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
	49	Morgan.Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
	53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL
	54	Duane60	2016-12-21 04:43:38	NULL	NULL	NULL	NULL
	57	Julien_Schmidt	2017-02-02 23:12:48	NULL	NULL	NULL	NULL
	66	Mike.Auer39	2016-07-01 17:36:15	NULL	NULL	NULL	NULL
	68	Franco_Keebler64	2016-11-13 20:09:27	NULL	NULL	NULL	NULL
	71	Nia_Haag	2016-05-14 15:38:50	NULL	NULL	NULL	NULL
	74	Hulda.Macejkovic	2017-01-25 17:17:28	NULL	NULL	NULL	NULL
	75	Leslie67	2016-09-21 05:14:01	NULL	NULL	NULL	NULL

The status bar at the bottom indicates 'Result 52 x'.



# Result

## Task 2- Identify users who have never posted a single photo on Instagram.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

ig\_clone

- Tables
  - comments
  - follows
  - likes
  - photo\_tags
  - photos
  - tags
  - users
- Views
- Stored Procedures
- Functions

sakila

sys

world

Administration Schemas

Information

No object selected

SQL File 1\* x

Limit to 1000 rows

```
39
40
41 • select * from users
42 left join photos on users.id = photos.user_id
43 where photos.user_id is null;
```

Result Grid

id	username	created_at	id	image_url	user_id	created_dat
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	Mckenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL
45	David.Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
49	Morgan.Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL
54	Duane60	2016-12-21 04:43:38	NULL	NULL	NULL	NULL
57	Julien_Schmidt	2017-02-02 23:12:48	NULL	NULL	NULL	NULL
66	Mike.Auer39	2016-07-01 17:36:15	NULL	NULL	NULL	NULL
68	Franco_Keebler64	2016-11-13 20:09:27	NULL	NULL	NULL	NULL
71	Nia_Haag	2016-05-14 15:38:50	NULL	NULL	NULL	NULL
74	Hulda.Macejkovic	2017-01-25 17:17:28	NULL	NULL	NULL	NULL
75	Leslie67	2016-09-21 05:14:01	NULL	NULL	NULL	NULL
76	Janelle.Nikolaus81	2016-07-21 09:26:09	NULL	NULL	NULL	NULL
80	Darby_Herzog	2016-05-06 00:14:21	NULL	NULL	NULL	NULL
81	Esther.Zulauf61	2017-01-14 17:02:34	NULL	NULL	NULL	NULL
83	Bartholome.Bernhard	2016-11-06 02:31:23	NULL	NULL	NULL	NULL
89	Jessyca_West	2016-09-14 23:47:05	NULL	NULL	NULL	NULL
90	Esmeralda.Mraz57	2017-03-03 11:52:27	NULL	NULL	NULL	NULL
91	Bethany20	2016-06-03 23:31:53	NULL	NULL	NULL	NULL

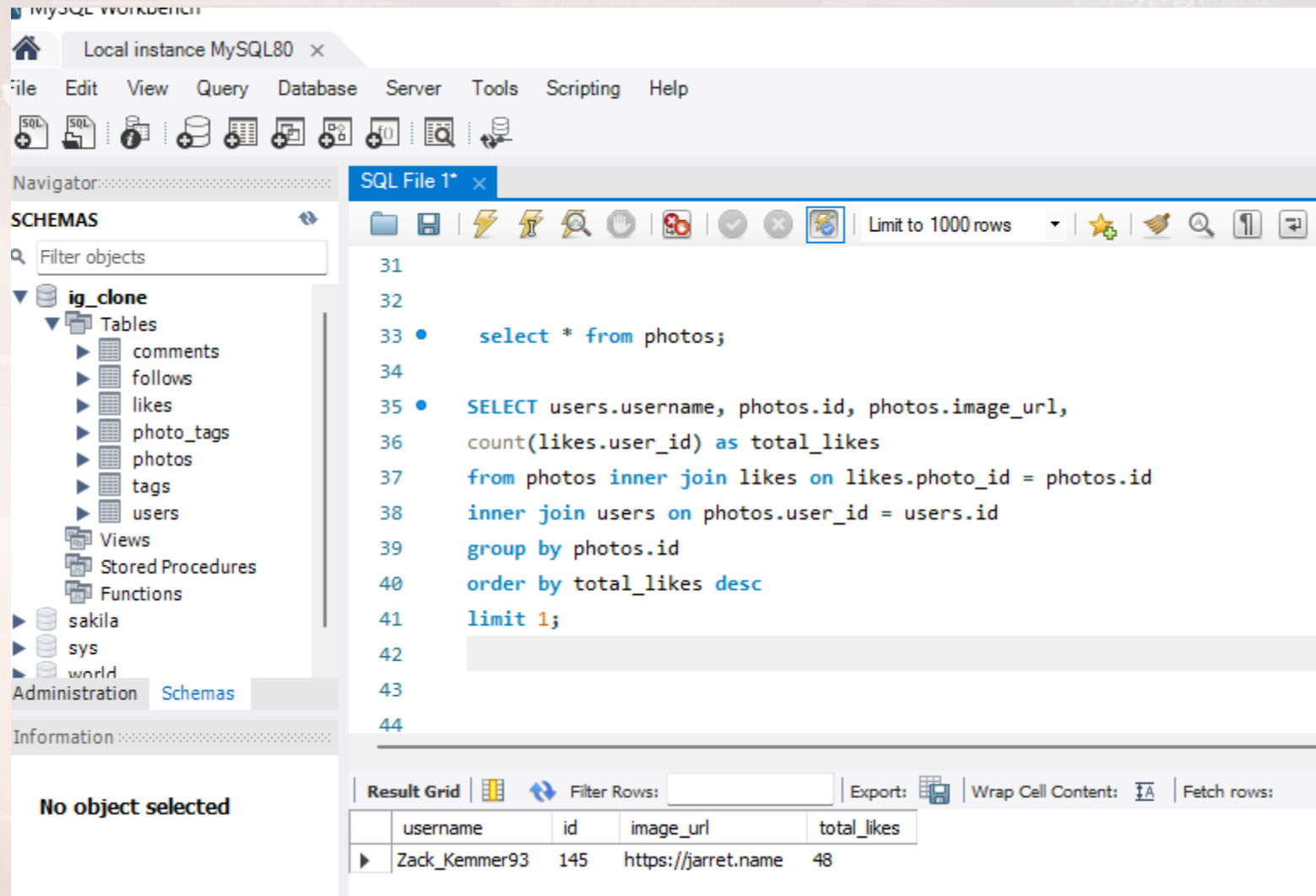
Result 52 x

A	B	C	D	E	F	G
id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	07/12/16 1:04	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	12/12/16 6:50	NULL	NULL	NULL	NULL
14	Jaclyn81	06/02/17 23:29	NULL	NULL	NULL	NULL
21	Rocio33	23/01/17 11:51	NULL	NULL	NULL	NULL
24	Maxwell.Halvorson	18/04/17 2:32	NULL	NULL	NULL	NULL
25	Tierra.Trantow	03/10/16 12:49	NULL	NULL	NULL	NULL
34	Pearl7	08/07/16 21:42	NULL	NULL	NULL	NULL
36	Ollie_Ledner37	04/08/16 15:42	NULL	NULL	NULL	NULL
41	Mckenna17	17/07/16 17:25	NULL	NULL	NULL	NULL
45	David.Osinski47	05/02/17 21:23	NULL	NULL	NULL	NULL
49	Morgan.Kassulke	30/10/16 12:42	NULL	NULL	NULL	NULL
53	Linnea59	07/02/17 7:49	NULL	NULL	NULL	NULL
54	Duane60	21/12/16 4:43	NULL	NULL	NULL	NULL
57	Julien_Schmidt	02/02/17 23:12	NULL	NULL	NULL	NULL
66	Mike.Auer39	01/07/16 17:36	NULL	NULL	NULL	NULL
68	Franco_Keebler64	13/11/16 20:09	NULL	NULL	NULL	NULL
71	Nia_Haag	14/05/16 15:38	NULL	NULL	NULL	NULL
74	Hulda.Macejkovic	25/01/17 17:17	NULL	NULL	NULL	NULL
75	Leslie67	21/09/16 5:14	NULL	NULL	NULL	NULL
76	Janelle.Nikolaus81	21/07/16 9:26	NULL	NULL	NULL	NULL
80	Darby_Herzog	06/05/16 0:14	NULL	NULL	NULL	NULL
81	Esther.Zulauf61	14/01/17 17:02	NULL	NULL	NULL	NULL
83	Bartholome.Bernhard	06/11/16 2:31	NULL	NULL	NULL	NULL
89	Jessyca_West	14/09/16 23:47	NULL	NULL	NULL	NULL
90	Esmeralda.Mraz57	03/03/17 11:52	NULL	NULL	NULL	NULL
91	Bethany20	03/06/16 23:31	NULL	NULL	NULL	NULL



# Result

**Task 3-** Determine the winner of the contest and provide their details to the team.



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'ig\_clone' with several tables: comments, follows, likes, photo\_tags, photos, tags, and users. The 'Information' pane at the bottom left shows 'No object selected'. The main editor window displays a SQL query in 'SQL File 1\*' that selects the top photo by total likes. The query is as follows:

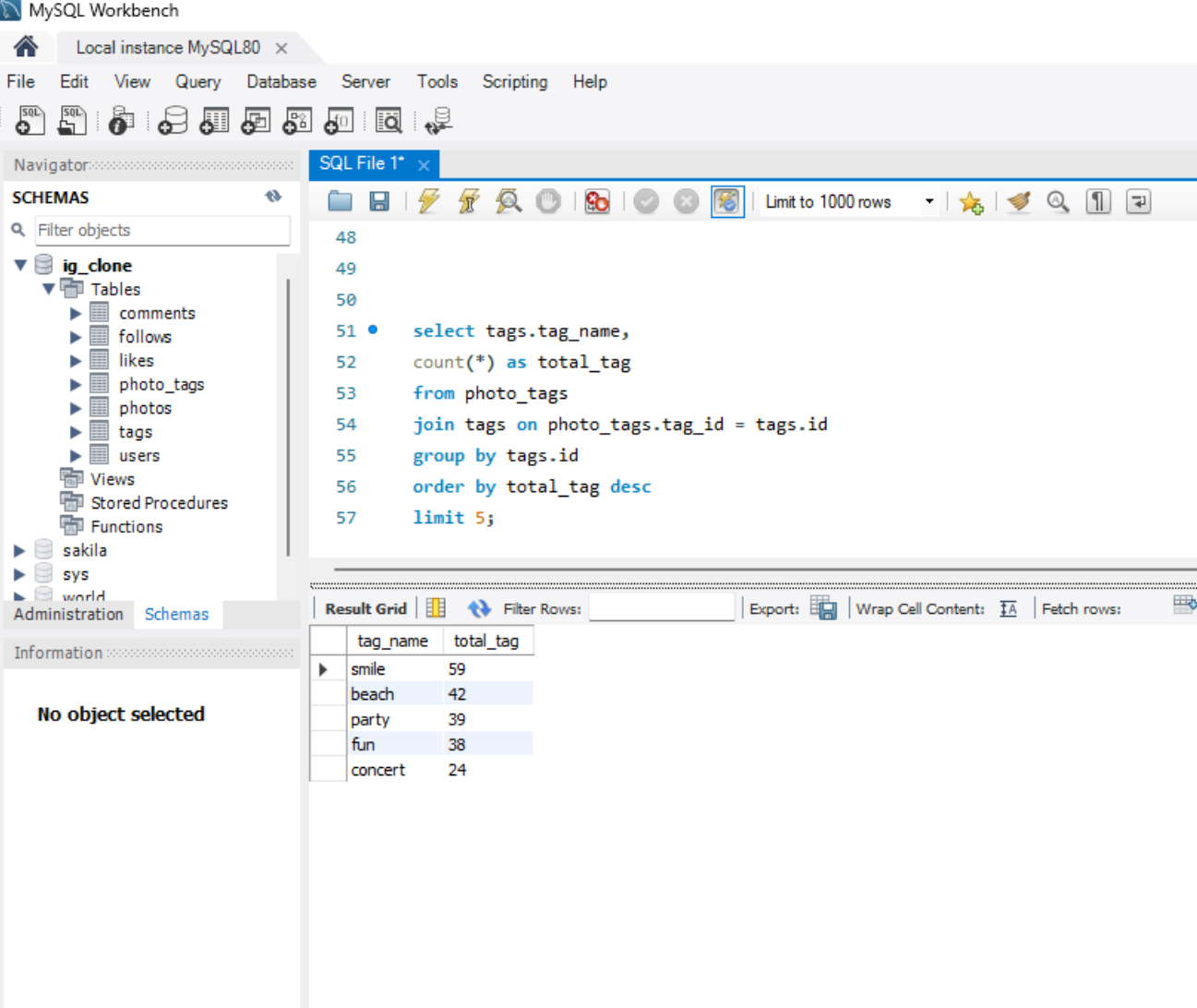
```
31
32
33 •   select * from photos;
34
35 •   SELECT users.username, photos.id, photos.image_url,
36       count(likes.user_id) as total_likes
37   from photos inner join likes on likes.photo_id = photos.id
38   inner join users on photos.user_id = users.id
39   group by photos.id
40   order by total_likes desc
41   limit 1;
42
43
44
```

Below the query editor, the 'Result Grid' shows the output of the query. It includes a table with the following data:

username	id	image_url	total_likes
Zack_Kemmer93	145	https://jarret.name	48

# Result

**Task 4-** Identify and suggest the top five most commonly used hashtags on the platform.



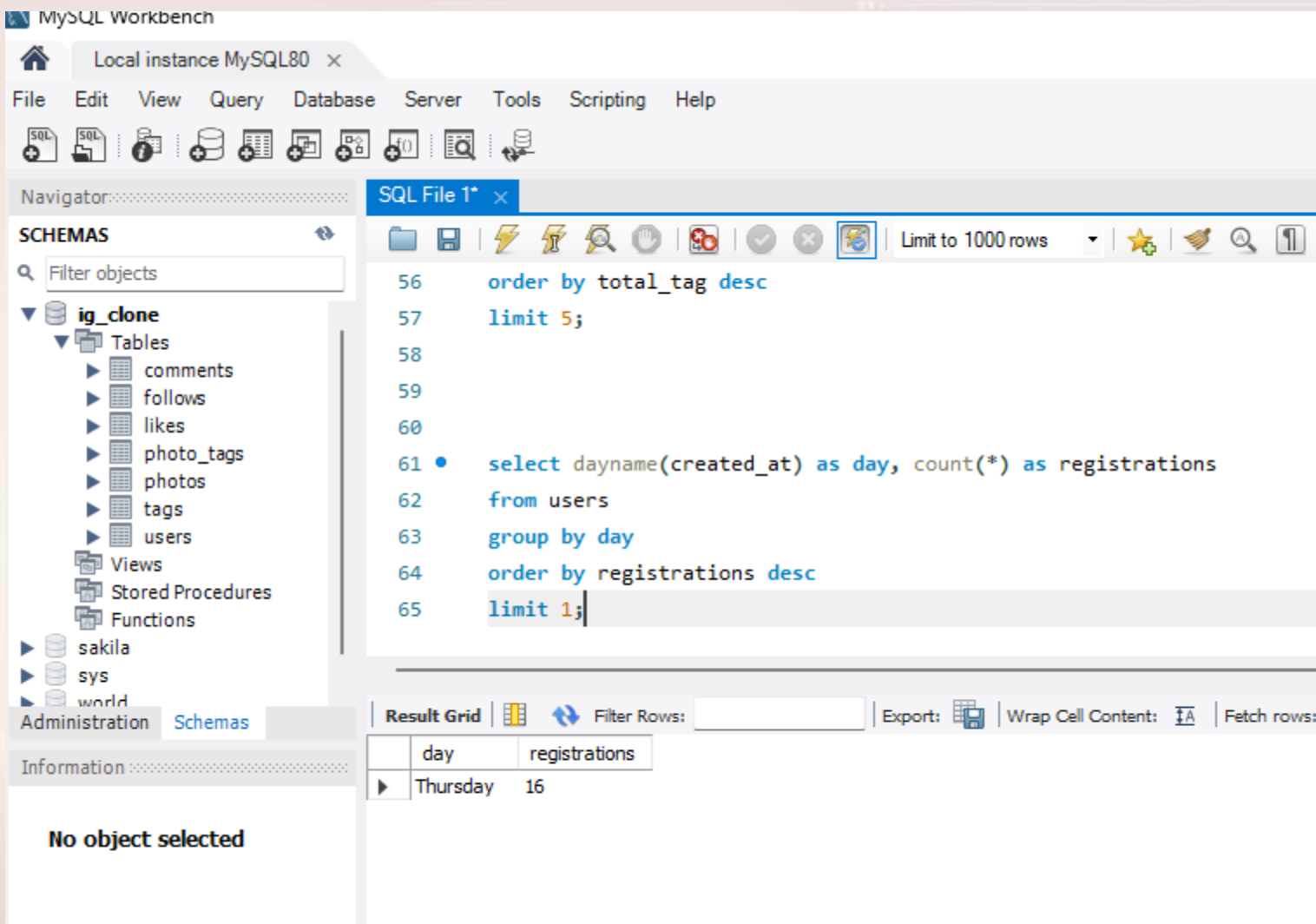
The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows the 'ig\_clone' database with tables like 'comments', 'follows', 'likes', 'photo\_tags', 'photos', 'tags', and 'users'. The main editor shows a SQL query to find the top 5 most common hashtags. The 'Result Grid' at the bottom shows the output of this query.

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

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

# Result

**Task 5-** 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 'ig\_clone' database, including tables like 'comments', 'follows', 'likes', 'photo\_tags', 'photos', 'tags', and 'users'. The main editor window shows a SQL query file with the following code:

```
56 order by total_tag desc
57 limit 5;
58
59
60
61 • select dayname(created_at) as day, count(*) as registrations
62 from users
63 group by day
64 order by registrations desc
65 limit 1;
```

Below the query editor, the 'Result Grid' tab is active, showing the following result:

day	registrations
Thursday	16

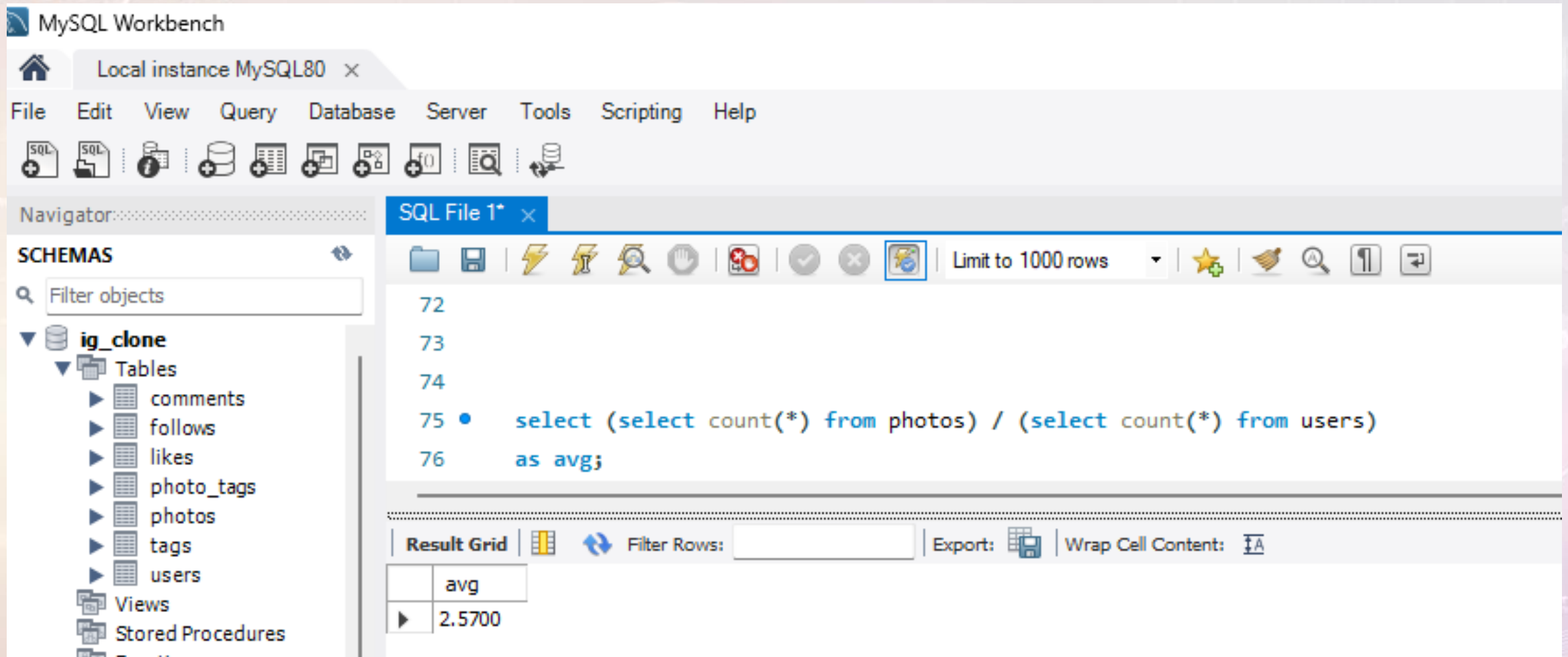
The bottom status bar indicates 'No object selected'.

Campaign should be scheduled on Thursday as it has the maximum no of engagements as compared to other days of the week.



# Result

**Task 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.



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'ig\_clone' with several tables: comments, follows, likes, photo\_tags, photos, tags, and users. The main editor window, titled 'SQL File 1', contains the following SQL query:

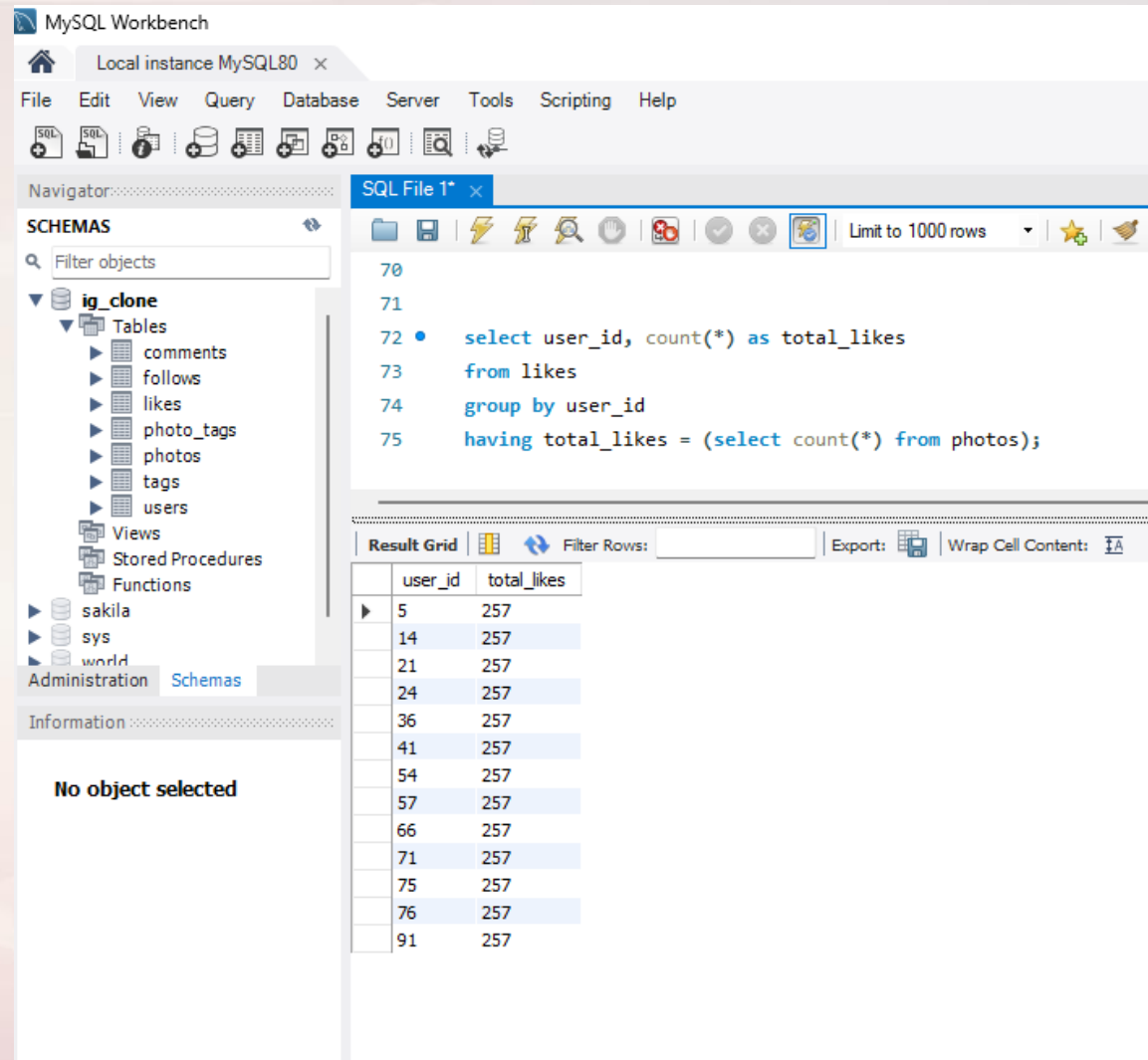
```
select (select count(*) from photos) / (select count(*) from users)
as avg;
```

Below the query editor, the 'Result Grid' is displayed, showing the result of the query:

	avg
	2.5700

# Result

**Task 7-** 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. On the left, the 'SCHEMAS' pane displays a tree view of the database structure for 'ig\_clone', including tables like 'comments', 'follows', 'likes', 'photo\_tags', 'photos', 'tags', and 'users'. The 'Query' tab is active, showing a SQL query that identifies users who have liked every photo. The 'Result Grid' at the bottom displays the query results, showing a list of user IDs and their total likes, all of which are 257.

```
70
71
72 • select user_id, count(*) as total_likes
73   from likes
74  group by user_id
75  having total_likes = (select count(*) from photos);
```

user_id	total_likes
5	257
14	257
21	257
24	257
36	257
41	257
54	257
57	257
66	257
71	257
75	257
76	257
91	257

# Conclusion

I have learned a lot of new concepts from the project, and will eventually help me in future development of my career.

