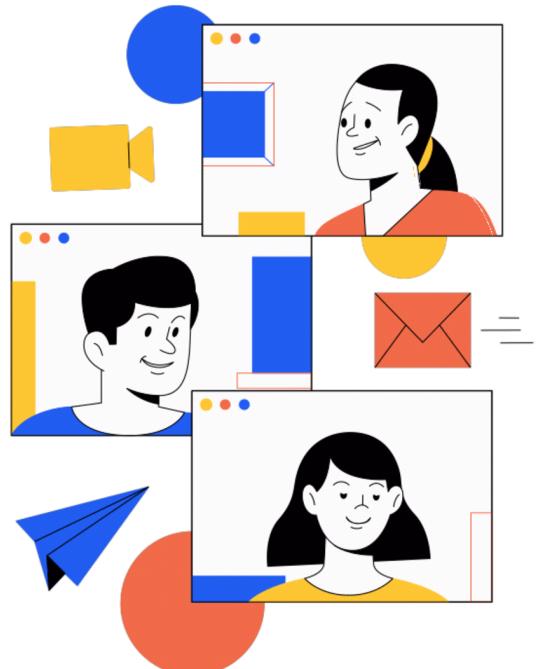


INSTAGRAM CLONE.....



ABOUT LEARNING.

- 1 Summary
- 2 Data Cleaning
- 3 query
- 4 Problem Solving
- 5 Insights



Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

instagram

Tables

tags

Columns

Indexes

Foreign Keys

Triggers

photos

photo_tags

likes

comments

tags

users

Columns

Administration Schemas

Information

Table: tags

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

username

Rodo33

Maxwell.Halvorson

Tierra.Trantow

Pearl2

Ollie_Ledner37

Mckenna17

David.Osinski97

Morgan.Kassulke

Linnea59

Duane60

Object Info Session Result 4

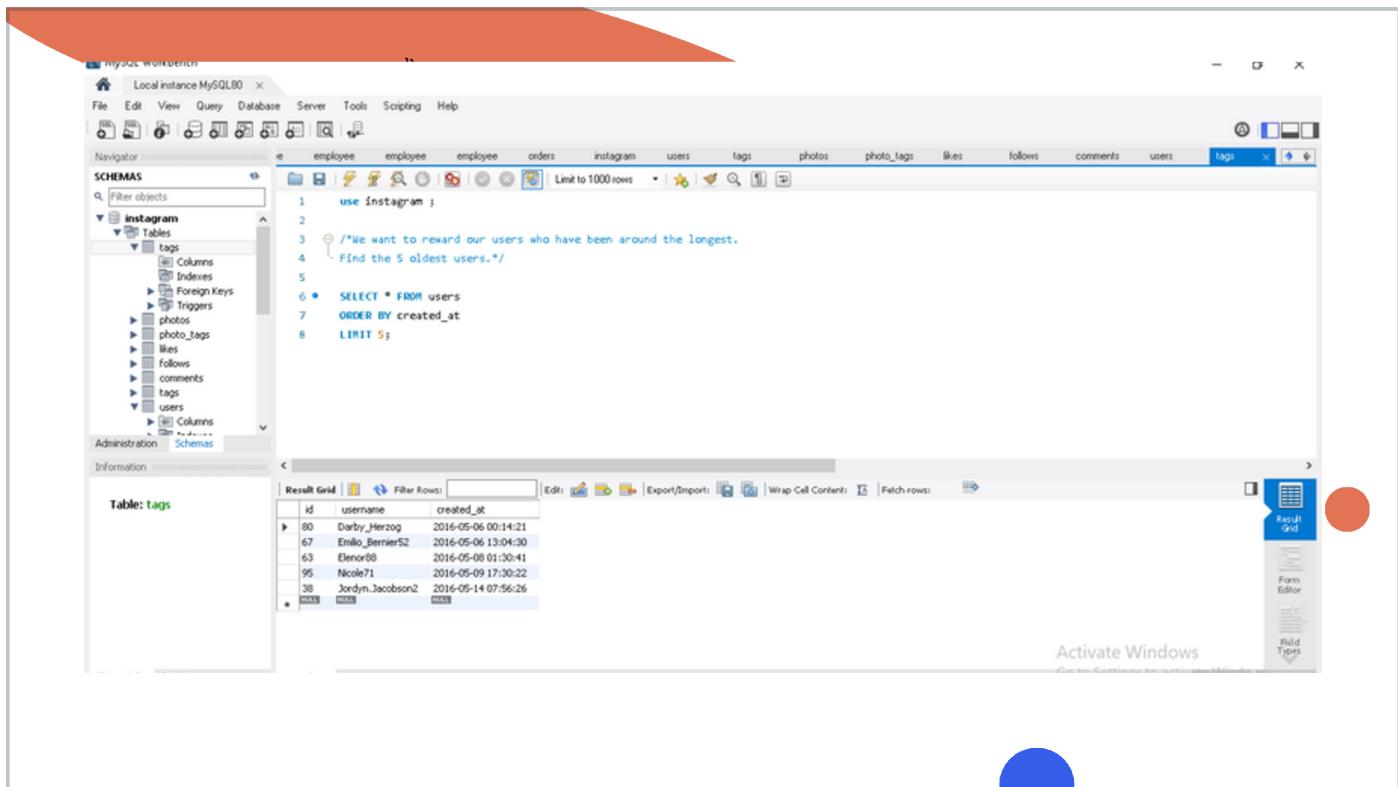
Activate Windows
Go to Settings to activate Windows

Result Grid Form Editor Field Types Read Only

```

11  /*We need to figure out when to schedule an ad campain*/
12 •  SELECT date_format(created_at,'N') AS 'day of the week', COUNT(*) AS 'total registration'
13  FROM users
14  GROUP BY 1
15  ORDER BY 2 DESC;
16
17  /*We want to target our inactive users with an email campaign.
18  Find the users who have never posted a photo*/
19 •  SELECT username
20  FROM users
21  LEFT JOIN photos ON users.id = photos.user_id
22  WHERE photos.id IS NULL
23
24

```

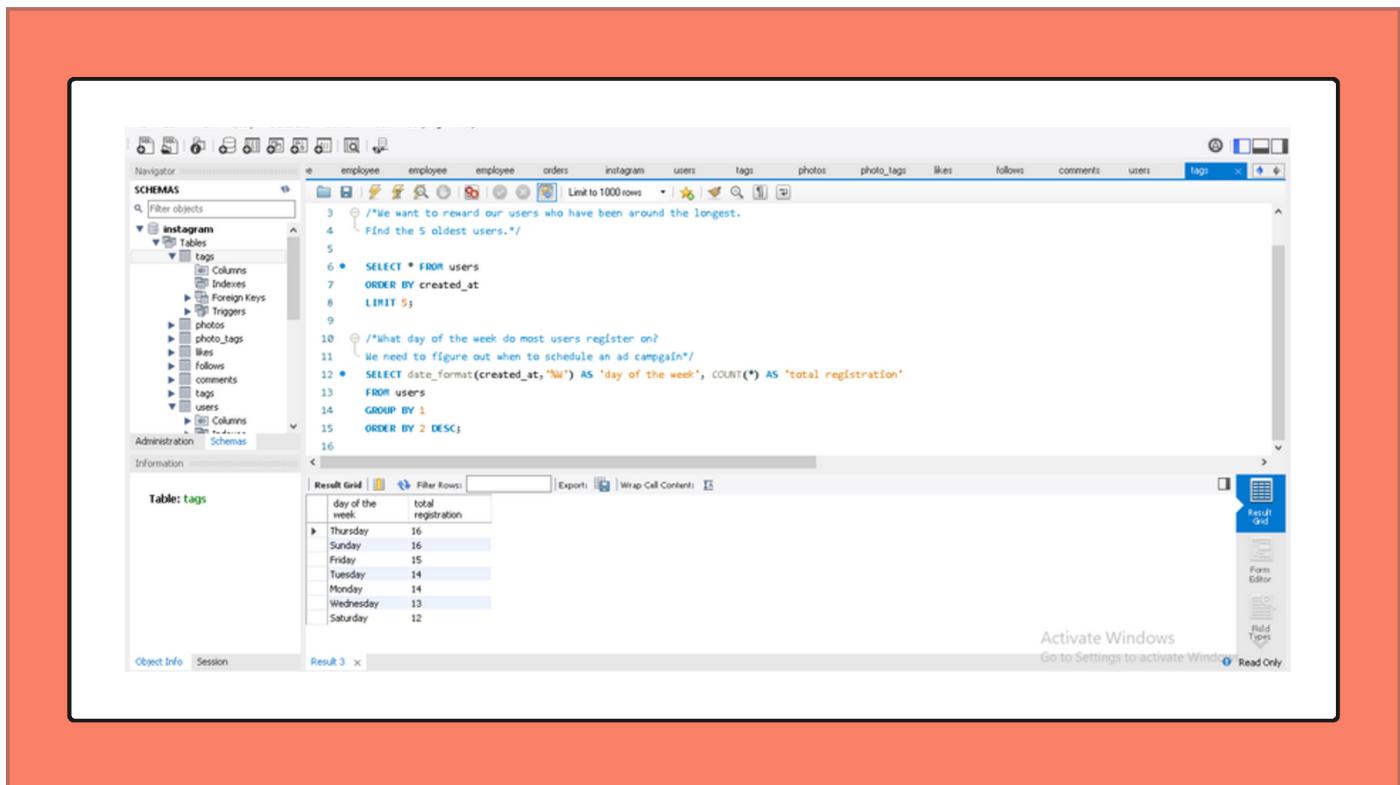


The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window has a 'Navigator' pane on the left showing the 'Schemas' tree, with 'tags' selected under the 'instagram' schema. The central pane displays a query editor with the following SQL code:

```
1 use instagram ;
2
3 /*We want to reward our users who have been around the longest.
4 Find the 5 oldest users.*/
5
6 * SELECT * FROM users
7 ORDER BY created_at
8 LIMIT 5;
```

The bottom pane shows a 'Result Grid' for the 'tags' table, which has three columns: id, username, and created_at. The data is as follows:

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Eleanor68	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn_Jacobson2	2016-05-14 07:56:26
58	Willie	2016-05-14 07:56:26

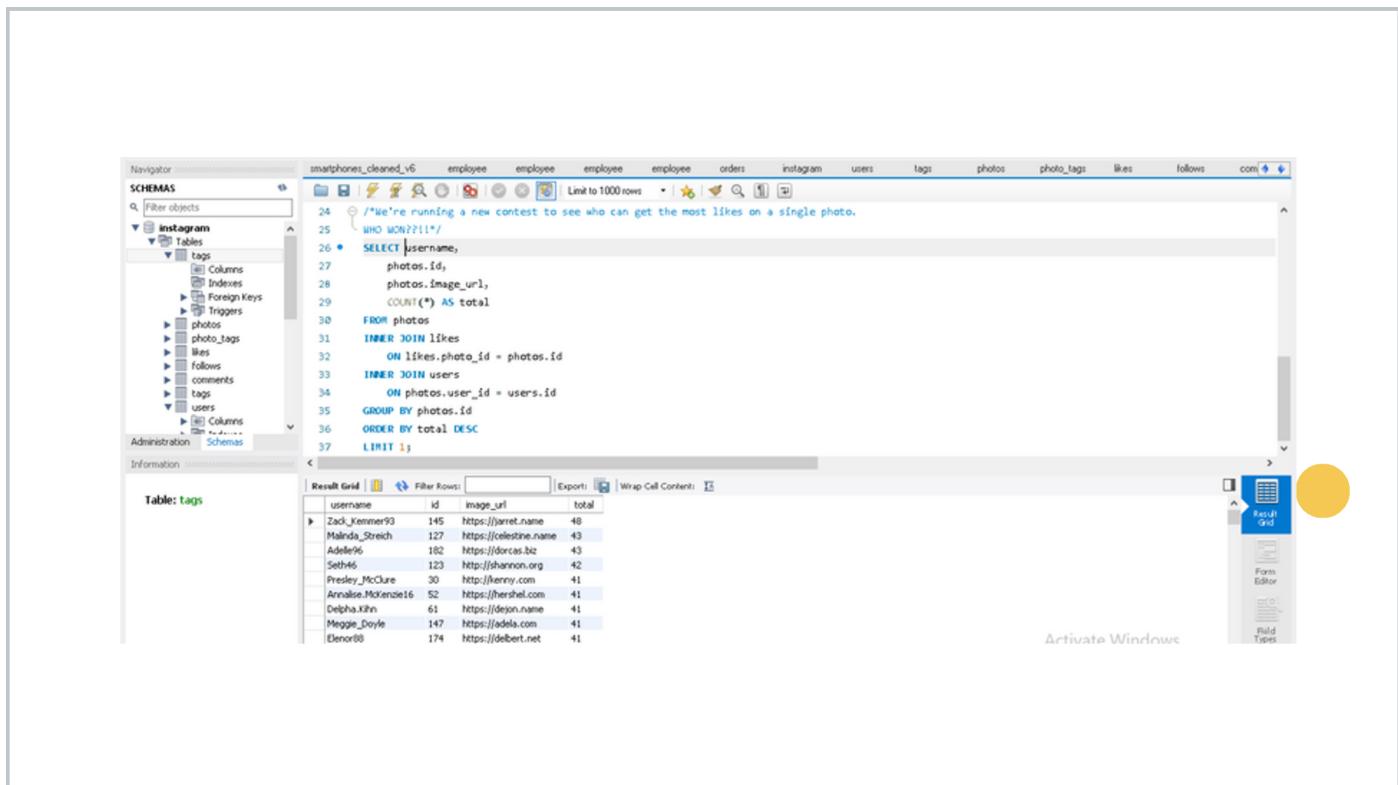


The screenshot shows a database management interface with the following details:

- Navigator:** On the left, it shows the **SCHEMAS** tree, which includes **employee**, **orders**, **instagram**, **users**, **tags**, and **photos**. The **tags** schema is currently selected.
- Query Editor:** The main area contains a SQL query:

```
3  /* We want to reward our users who have been around the longest.
4  Find the 5 oldest users.*/
5
6  • SELECT * FROM users
7  ORDER BY created_at
8  LIMIT 5;
9
10 /* What day of the week do most users register on?
11 We need to figure out when to schedule an ad campaign*/
12 • SELECT date_format(created_at,"%W") AS 'day of the week', COUNT(*) AS 'total registration'
13  FROM users
14  GROUP BY 1
15  ORDER BY 2 DESC;
16
```
- Results Grid:** Below the query editor, the results for the second query are displayed in a grid:

day of the week	total registration
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12
- Toolbars and Buttons:** The top bar includes standard database navigation buttons like back, forward, search, and refresh, along with a "Limit to 1000 rows" dropdown.
- Information Bar:** At the bottom, it shows "Table: tags", "Result 3", and a "Read Only" status indicator.



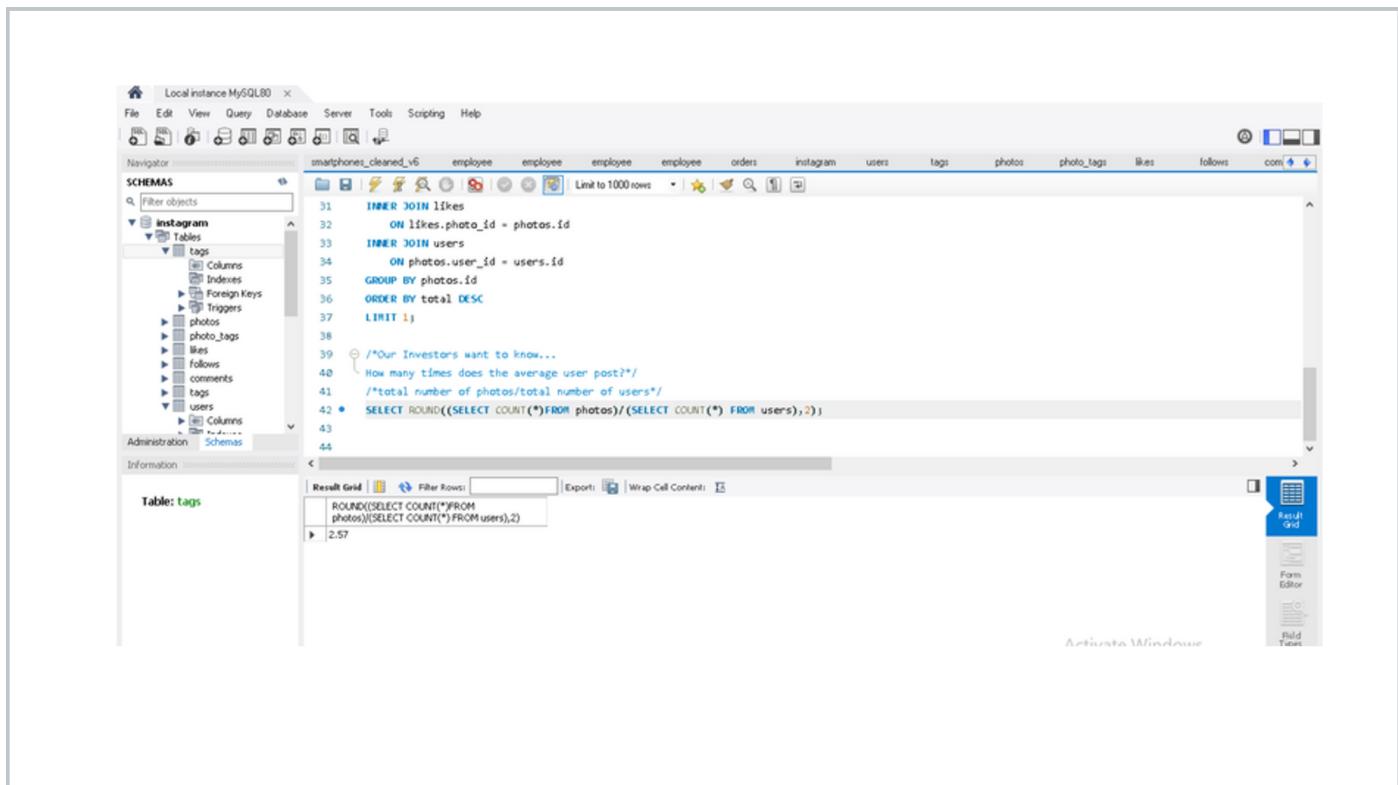
The screenshot shows a database management interface with the following details:

- Navigator:** Shows the schema structure for the `smartphones_cleaned_v6` database, including tables like `employee`, `orders`, `tags`, `users`, and `photos`.
- Query Editor:** Displays a SQL query to find the user with the most likes on a single photo:


```

24  /*We're running a new contest to see who can get the most likes on a single photo.
25  WHO WON??*/
26  SELECT username,
27    photos.id,
28    photos.image_url,
29    COUNT(*) AS total
30  FROM photos
31  INNER JOIN likes
32    ON likes.photo_id = photos.id
33  INNER JOIN users
34    ON photos.user_id = users.id
35  GROUP BY photos.id
36  ORDER BY total DESC
37  LIMIT 1;
      
```
- Result Grid:** Shows the results of the query in a grid format:

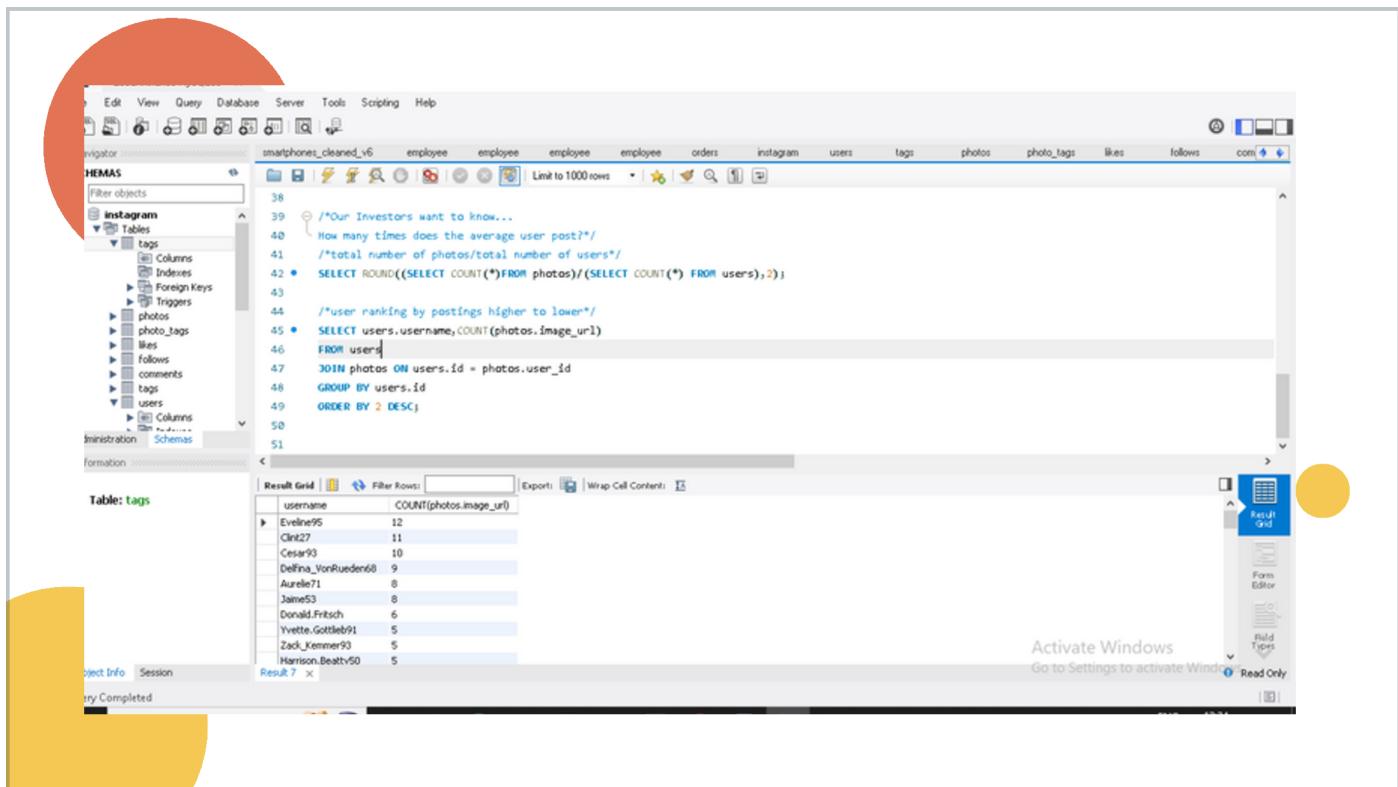
username	id	image_url	total
Zach_Kemmer93	145	https://jarret.name	48
Malinda_Streich	127	https://celesteine.name	43
Adelle96	182	https://dorcas.biz	43
Seth46	123	http://shannon.org	42
Presley_McCure	30	http://jenny.com	41
Annalise.Mckenzie16	52	https://hershel.com	41
Delpha_Xhnn	61	https://dejon.name	41
Meggie_Doyle	147	https://adela.com	41
Elenor08	174	https://debert.net	41
- Toolbar:** Includes buttons for Result Grid, Form Editor, and Field Types.
- Information:** Shows the table structure for the `tags` table.



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar, titled 'Navigator', shows the 'Schemas' section with 'smartphones_cleaned_v6' selected. Under 'Tables', the 'tags' table is expanded, showing its columns: id, name, and photo_id. The main workspace contains a query editor with the following SQL code:

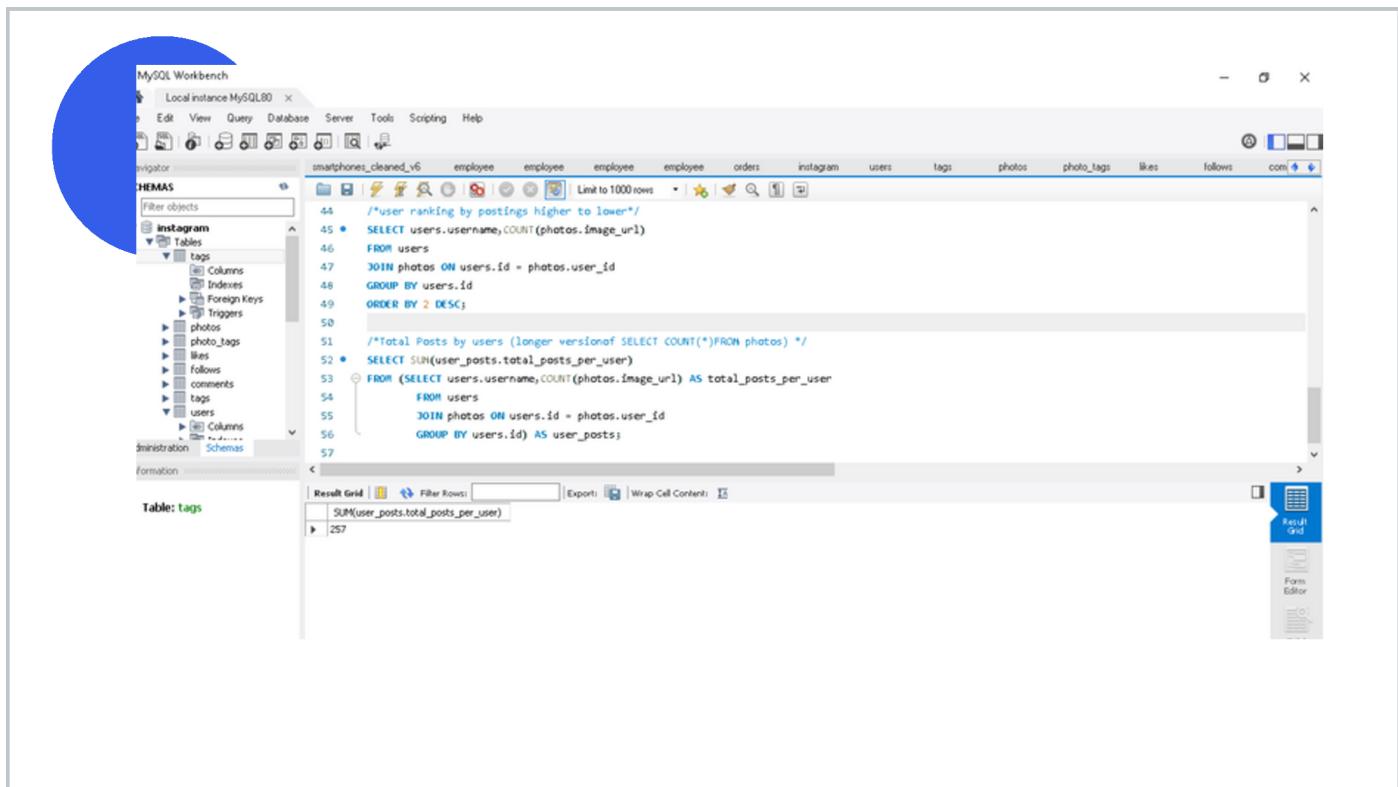
```
31 INNER JOIN likes
32     ON likes.photo_id = photos.id
33 INNER JOIN users
34     ON photos.user_id = users.id
35 GROUP BY photos.id
36 ORDER BY total DESC
37 LIMIT 1
38
39 /*Our Investors want to know...
40 - How many times does the average user post?*/
41 /*total number of photos/total number of users*/
42 • SELECT ROUND((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users),2)
43
44
```

The results grid shows a single row with the value 2.57. The bottom right corner of the interface has a toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Tables'.



/*Our Investors want to know...
How many times does the average user post?*/
SELECT ROUND((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users),2);
/*user ranking by postings higher to lower*/
SELECT users.username,COUNT(photos.image_url)
FROM users
JOIN photos ON users.id = photos.user_id
GROUP BY users.id
ORDER BY 2 DESC;

username	COUNT(photos.image_url)
Eveline95	12
Clint27	11
Cesar93	10
Delfina_VonRueden68	9
Auriele71	8
Jaime53	8
Donald_Fritsch	6
Yvette_Gottlieb91	5
Zack_Kemmer93	5
Harrison_Heatty50	5



MySQL Workbench

Local instance MySQL800

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Instagram

Tables

tags

Columns

Indexes

Foreign Keys

Triggers

photos

photo_tags

likes

comments

tags

users

Columns

Administration Schemas

Formation

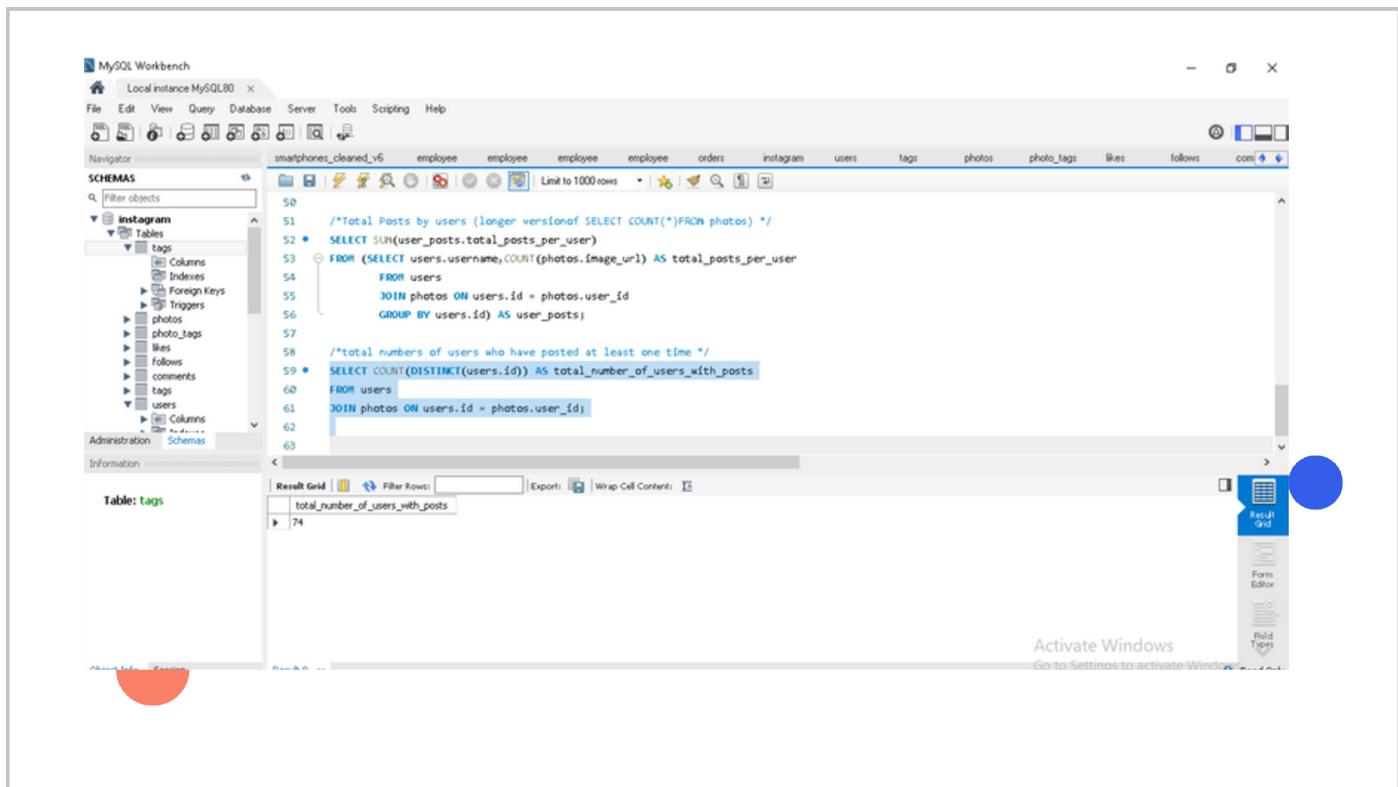
Table: tags

Result Grid | Filter Rows! | Export! | Wrap Cell Content: 257

44 /*user ranking by postings higher to lower*/
45 • SELECT users.username,COUNT(photos.image_url)
46 FROM users
47 JOIN photos ON users.id = photos.user_id
48 GROUP BY users.id
49 ORDER BY 2 DESC;
50
51 /*Total Posts by users (longer version of SELECT COUNT(*)FROM photos) */
52 • SELECT SUM(user_posts.total_posts_per_user)
53 FROM (SELECT users.username,COUNT(photos.image_url) AS total_posts_per_user
54 FROM users
55 JOIN photos ON users.id = photos.user_id
56 GROUP BY users.id) AS user_posts;
57

Result Grid

Form Editor



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Instagram

Tables

tags

Columns

Indexes

Foreign Keys

Triggers

photos

photo_tags

likes

tags

users

Columns

Administration Schemas

Information

Table: tags

smartphones_cleaned_v6 employee employee employee employee orders instagram users tags photos photo_tags likes follows com

50 /*Total Posts by users (longer version of SELECT COUNT(*) FROM photos) */
51 SELECT SUM(user_posts.total_posts_per_user) AS total_posts_per_user
52 FROM (SELECT users.username, COUNT(photos.image_url) AS total_posts_per_user
53 FROM users
54 JOIN photos ON users.id = photos.user_id
55 GROUP BY users.id) AS user_posts
56
57 /*total numbers of users who have posted at least one time */
58 SELECT COUNT(DISTINCT(users.id)) AS total_number_of_users_with_posts
59 FROM users
60 JOIN photos ON users.id = photos.user_id
61
62
63

Result Grid | Filter Rows: Export: Wrap Cell Content: 74

total_number_of_users_with_posts

Activate Windows
Go to Settings to activate Windows

Result Grid Form Editor Field Types

SCHEMAS

Filter objects

instagram

Tables

tags

Columns

Indexes

Foreign Keys

Triggers

photos

photo_tags

files

comments

tags

users

Columns

Administration Schemas

Information

Table: tags

Result Grid | Filter Rows: Export: Wrap Cell Content: 15

Result Grid Form Editor Field Types

Activate Windows Go to Settings to activate Windows Read Only

Result 10 X

```

57
58  /*total numbers of users who have posted at least one time */
59  • SELECT COUNT(DISTINCT(users.id)) AS total_number_of_users_with_posts
60  FROM users
61  JOIN photos ON users.id = photos.user_id;
62
63  /*A brand wants to know which hashtags to use in a post
64  What are the top 5 most commonly used hashtags?*/
65  • SELECT tag_name, COUNT(tag_name) AS total
66  FROM tags
67  JOIN photo_tags ON tags.id = photo_tags.tag_id
68  GROUP BY tags.id
69  ORDER BY total DESC;
70

```

tag_name	total
smile	59
beach	42
party	39
fun	38
food	24
lol	24
concert	24
hair	23
happy	22
beauty	20

Query Completed

Turn here to search

Activate Windows Go to Settings to activate Windows Read Only

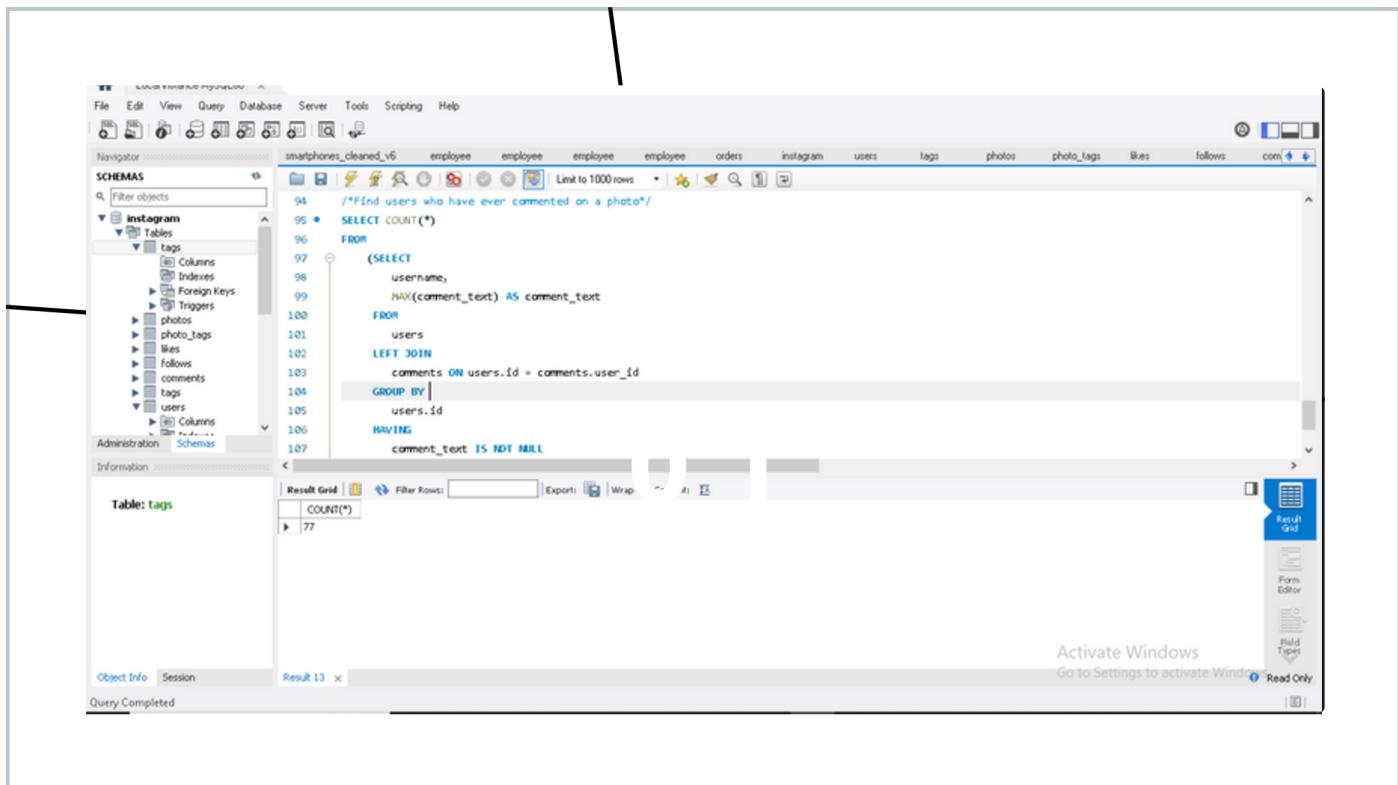
12:36

```

71 /*We have a small problem with bots on our site...
72 Find users who have liked every single photo on the site*/
73 SELECT users.id,username, COUNT(users.id) As total_likes_by_user
74 FROM users
75 JOIN likes ON users.id = likes.user_id
76 GROUP BY users.id
77 HAVING total_likes_by_user = (SELECT COUNT(*) FROM photos);
78
79 /*We also have a problem with celebrities
80 Find users who have never commented on a photo*/
81 SELECT username,comment_text
82 FROM users
83 LEFT JOIN comments ON users.id = comments.user_id

```

id	username	total_likes_by_user
5	Annya_Hackett	257
14	Jacyln81	257
21	Rico03	257
24	Maxwell_Halvorsen	257
36	Ollie_Ledner37	257
41	McKenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike_Auer39	257
71	Na_Haag	257



The screenshot shows a database management interface with the following details:

- File Edit View Query Database Server Tools Scripting Help**
- Navigator** pane on the left showing the **SCHEMAS** tree, with **Instagram** selected, and the **Tables** tree under it, showing **tags** as the selected table.
- Query Editor** pane in the center displaying the following SQL query:

```
94  /*Find users who have ever commented on a photo*/
95  •  SELECT COUNT(*)
96  FROM
97  (SELECT
98      username,
99      MAX(comment_text) AS comment_text
100     FROM
101     users
102     LEFT JOIN
103     comments ON users.id = comments.user_id
104     GROUP BY
105         users.id
106     HAVING
107         comment_text IS NOT NULL
```

The result grid shows a single row with the value 77.

Bottom status bar: **Object Info Session Result 13 x Query Completed**

Right side: **Activate Windows**, **Go to Settings to activate Windows**, **Read Only**

SWOT

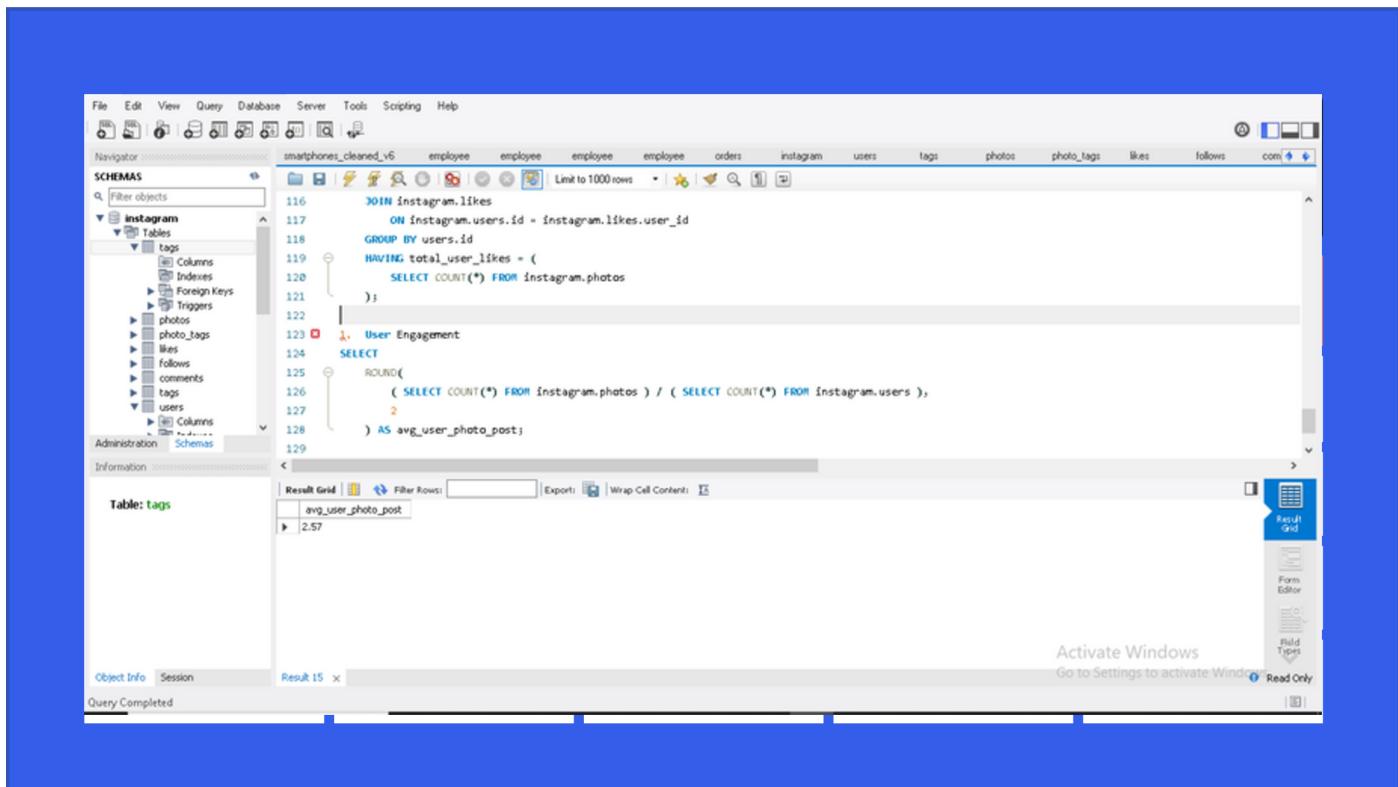
```
108     ) AS total_number_users_with_comments;
109
110  2. Bot Accounts
111  SELECT
112      instagram.users.id AS user_id,
113      instagram.users.username,
114      COUNT(*) AS total_user_likes
115  FROM instagram.users
116  JOIN instagram.likes
117      ON instagram.users.id = instagram.likes.user_id
118  GROUP BY users.id
119  HAVING total_user_likes = (
120      SELECT COUNT(*) FROM instagram.photos
121  )
```

user_id	username	total_user_likes
5	Aniya_Hackett	257
14	Jadyn81	257
21	Rod033	257
24	MaxwellHalvorson	257
36	Ollie_Ledner37	257
41	McKenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike_Auer39	257
71	Nia_Haas	257

Result 14

Activate Windows
Go to Settings to activate Windows

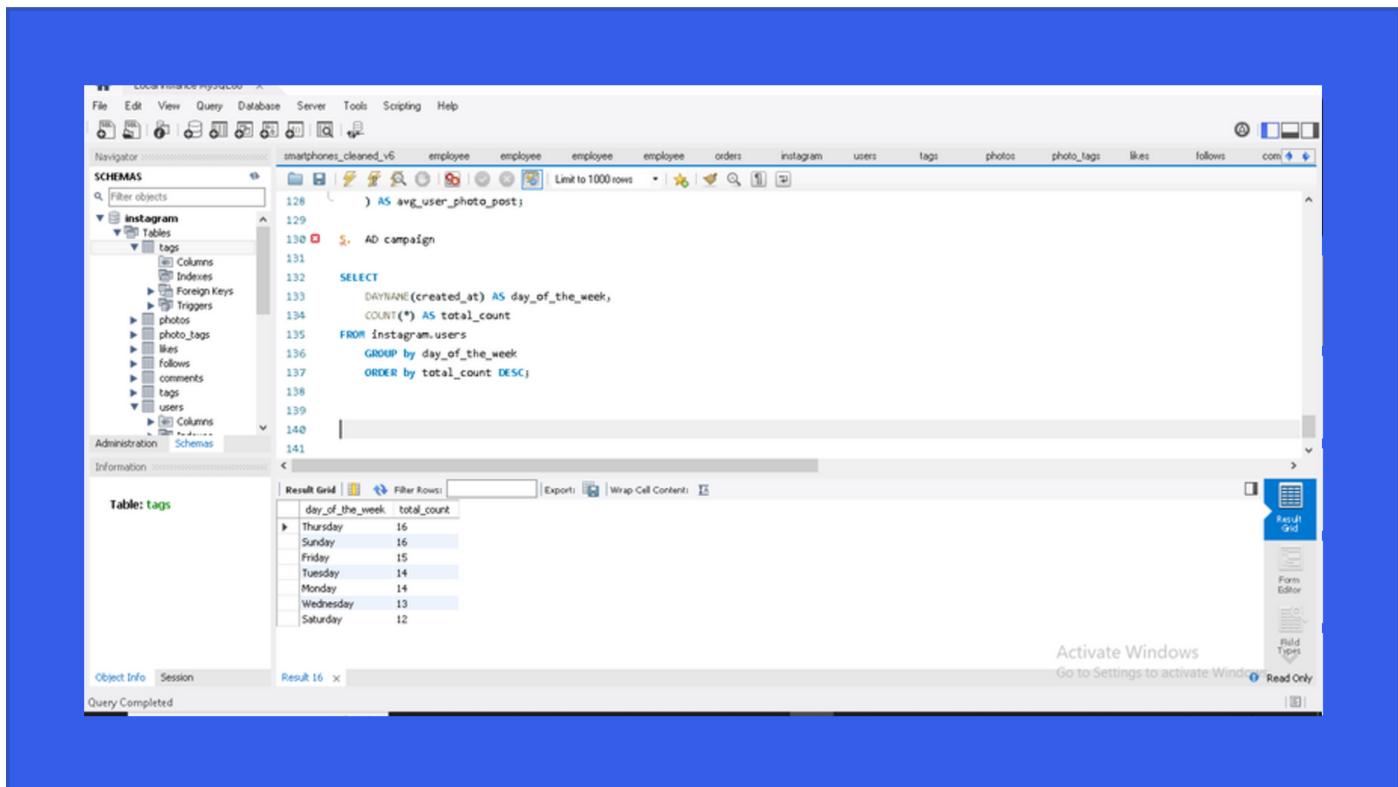
Result Grid Form Editor Field Types Read Only



The screenshot shows a database query tool interface with the following details:

- File Edit View Query Database Server Tools Scripting Help** (Menu Bar)
- Navigator** (Left Panel): Shows the **smartphones_cleaned_v6** database with the **instagram** schema selected. Under **Tables**, the **tags** table is expanded, showing its columns: **id**, **name**, **photo_id**, **user_id**, and **post_id**.
- Query Editor** (Main Area): Displays the following SQL query:

```
116     JOIN instagram.likes
117         ON instagram.users.id = instagram.likes.user_id
118     GROUP BY users.id
119     HAVING total_user_likes = (
120         SELECT COUNT(*) FROM instagram.photos
121     );
122
123 1. User Engagement
124  SELECT
125     ROUND(
126         ( SELECT COUNT(*) FROM Instagram.photos ) / ( SELECT COUNT(*) FROM Instagram.users ),
127         2
128     ) AS avg_user_photo_post;
129
```
- Result Grid** (Bottom): Shows the result of the query, which is a single row with the value **2.57** for the column **avg_user_photo_post**.
- Information** (Bottom Left): Shows the **tags** table with the value **2.57**.
- Object Info Session Result 15 x** (Bottom Left): Shows the status **Query Completed**.
- Activate Windows** (Bottom Right): A message to activate Windows.
- Result Grid** (Bottom Right): A button to switch to the result grid view.



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window has a 'Navigator' pane on the left containing a 'SCHEMAS' tree with 'smartphones_cleaned_v6' selected, and a 'Tables' tree for the 'tags' table. The central pane displays a query editor with the following SQL code:

```
128 ) AS avg_user_photo_post;
129
130 ② AD campaign
131
132 SELECT
133   DAYNAME(created_at) AS day_of_the_week,
134   COUNT(*) AS total_count
135   FROM Instagram.users
136   GROUP BY day_of_the_week
137   ORDER BY total_count DESC;
138
139
140
141
```

The results of the query are shown in a 'Result Grid' table:

day_of_the_week	total_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

At the bottom of the interface, there are tabs for 'Object Info', 'Session', and 'Result 16'. A message 'Activate Windows' with a link 'Go to Settings to activate Windows' is visible. The status bar at the bottom left says 'Query Completed'.

ALL QUESTION

Find the 5 oldest users

What day of the week do most users register on?

Find the users who have never posted a photo

We're running a new contest to see who can get the most likes on a single photo.

\

How many times does the average user post?
user ranking by postings higher to lower

Total Posts by users (longer version of SELECT
COUNT(*) FROM photos

total numbers of users who have posted at least
one time

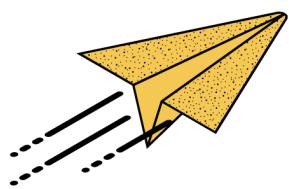
What are the top 5 most commonly used hashtags?

Find users who have liked every single photo on
the site

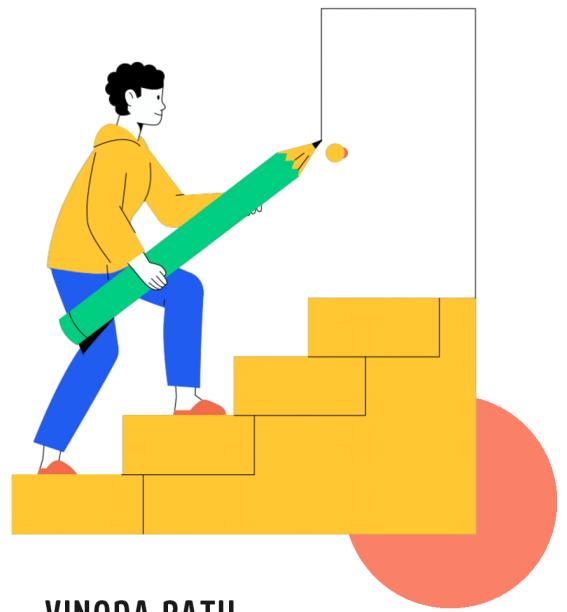
Find users who have never commented on a
photo

RESOURCE PAGE





THANK YOU!



VINODA PATIL