

# Project-2 Instagram User Analytics

## **Project Description :**

This Project is an attempt to derive business insights for marketing, product & development teams based on the insights made out of the queries asked by the management team.

## **The following queries were :**

1. People who have been using platform for the longest time ?
2. Who have never posted a single photo on Instagram ?
3. Identify the winner of the contest who has got the most likes for a photo ?
4. Identify the top 5 most commonly used hashtags on the platform ?
5. Which is the best day to launch Ad campaign ?
6. Provide how many times does average user posts on Instagram and the total number of photos on Instagram and total number of users ?
7. Provide data on users (bots) who have liked every single photo on the site.

## **Approach :**

Created a database using MySQL that can store the data and support the queries that needs to be processed for analyzing and loaded the data into the database and used MySQL queries to perform the analysis and made insights about the queries that were asked by the management team.

## **Tech-Stack Used :**

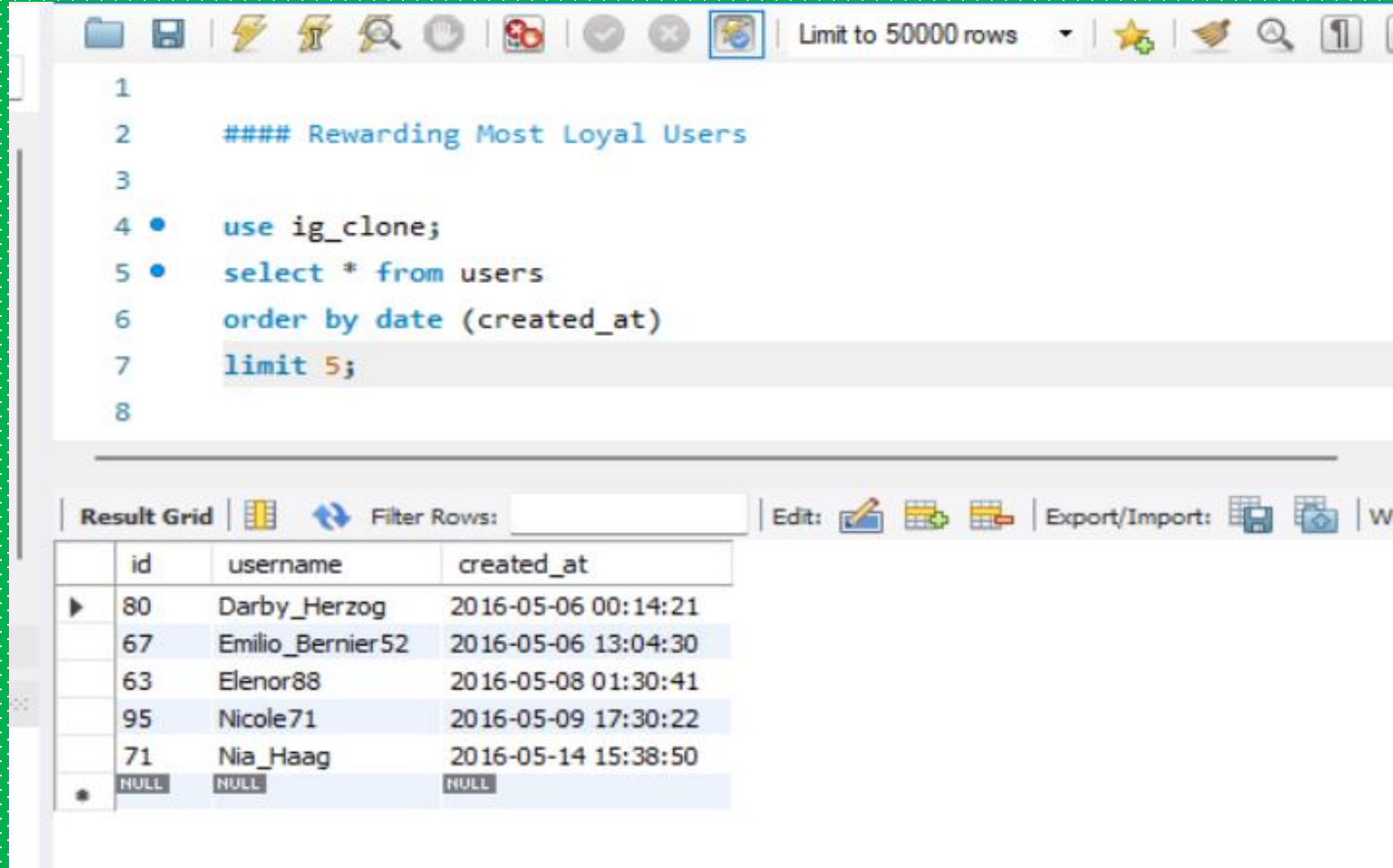
1. MySQL - To run the Query and extract data from database.
2. MS Excel – To Store the Extracted tables from the database.
3. MS PPT – To prepare the report based the insights from the database.

## **Insights :**

The following were the SQL queries and their outputs for the queries that were asked by the management team.

## A) Marketing:

### 1. Rewarding Most Loyal Users:



The screenshot shows a SQL query editor interface. The query is as follows:

```
1  
2  #### Rewarding Most Loyal Users  
3  
4  • use ig_clone;  
5  • select * from users  
6    order by date (created_at)  
7    limit 5;  
8
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has columns for id, username, and created\_at. The results are 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	Elenor88	2016-05-08 01:30:41
	95	Nicole71	2016-05-09 17:30:22
	71	Nia_Haag	2016-05-14 15:38:50
•	NULL	NULL	NULL

They are the 5 oldest users of the Instagram from the database provided.

## 2. Remind Inactive Users to Start Posting:

```
8
9   ### Remind Inactive Users to Start Posting
10 • select * from users
11   left join photos
12   on users.id = photos.user_id
13   where user_id is null;
14
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	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	Rocio33	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

Result 5 x

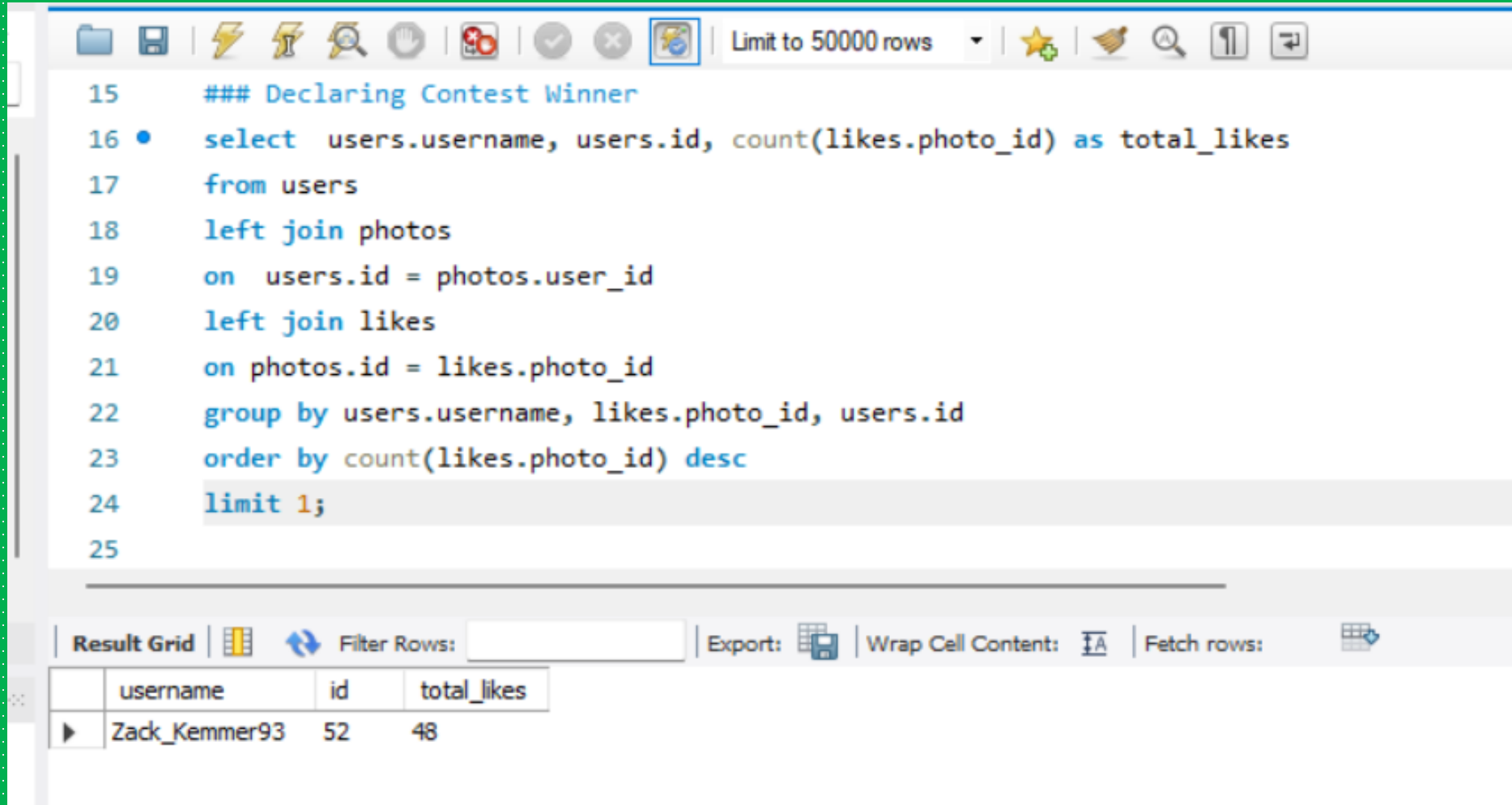
```
8
9   ### Remind Inactive Users to Start Posting
10 • select * from users
11   left join photos
12   on users.id = photos.user_id
13   where user_id is null;
14
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	id	username	created_at	id	image_url	user_id	created_at
	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

These are the users who have not yet posted anything on Instagram.

### 3. Declaring Contest Winner:



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The SQL editor contains the following query:

```
15  ### Declaring Contest Winner
16  •  select  users.username, users.id, count(likes.photo_id) as total_likes
17      from users
18      left join photos
19      on  users.id = photos.user_id
20      left join likes
21      on photos.id = likes.photo_id
22      group by users.username, likes.photo_id, users.id
23      order by count(likes.photo_id) desc
24      limit 1;
25
```





Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. The result grid displays the following data:

	username	id	total_likes
▶	Zack_Kemmer93	52	48


Zack\_Kemmer93 is the winner with most number of likes for a single photo.

## 4. Hashtag Researching :

```
25
26   ### Top 5 Hashtag Researching
27   •   select tag_name as "TOP 5 Tags", count(tags.id) as "No. of Times Used"
28         from photos
29         left join photo_tags
30         on photos.id = photo_tags.photo_id
31         left join tags
32         on photo_tags.tag_id = tags.id
33         Group by tags.id
34         order by count(id) desc
35         limit 5;
```

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




	TOP 5 Tags	No. of Times Used
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

Result 30 x 

The list of all the hashtags were the most used in Instagram.

## 5. Launch AD Campaign :

```
36
37   ### Best Day for Launch of AD Campaign
38 • select dayname(created_at) as Days_of_the_week, count(*) as Total
39   from users
40   group by Days_of_the_week
41   order by Total desc
42
43
44
--
```

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

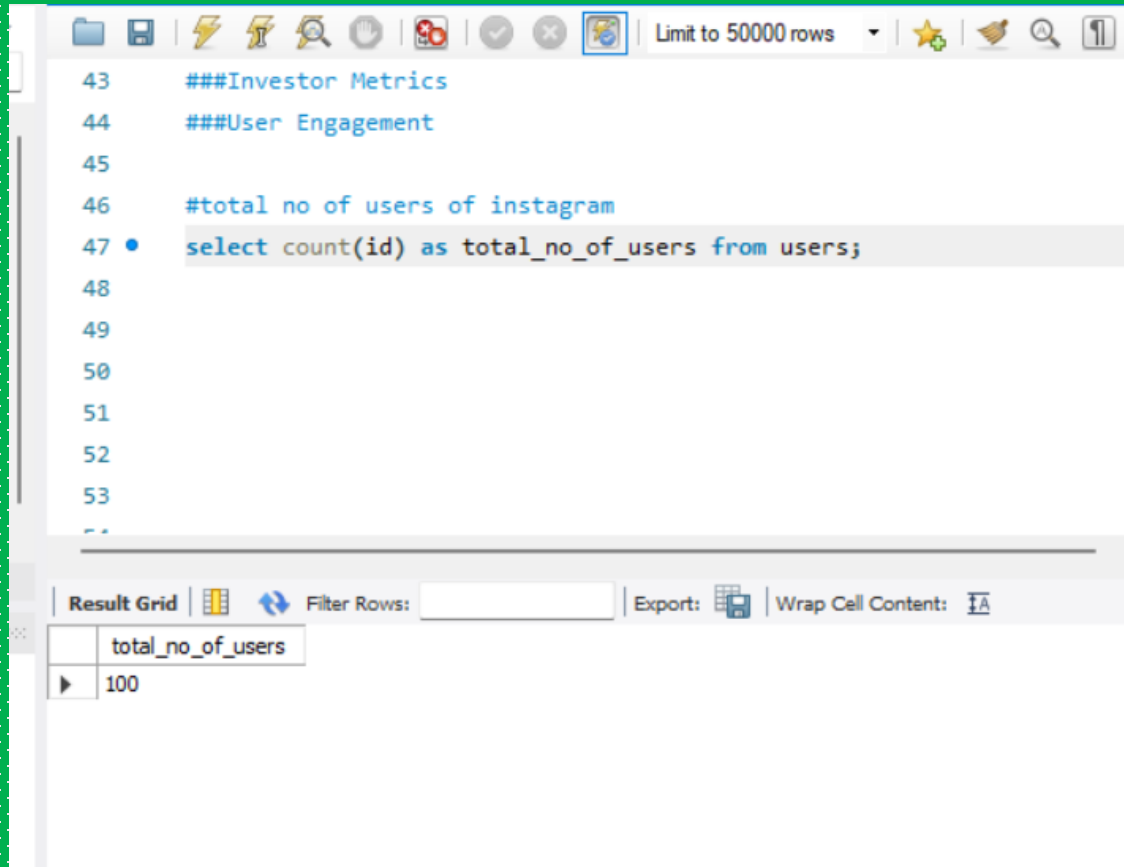
	Days_of_the_week	Total
▶	Thursday	16
	Sunday	16
	Friday	15
	Tuesday	14
	Monday	14
	Wednesday	13
	Saturday	12

Result 34 x

These days of the week were the most users registered on.

## B) Investor Metrics

### 6. User Engagement:



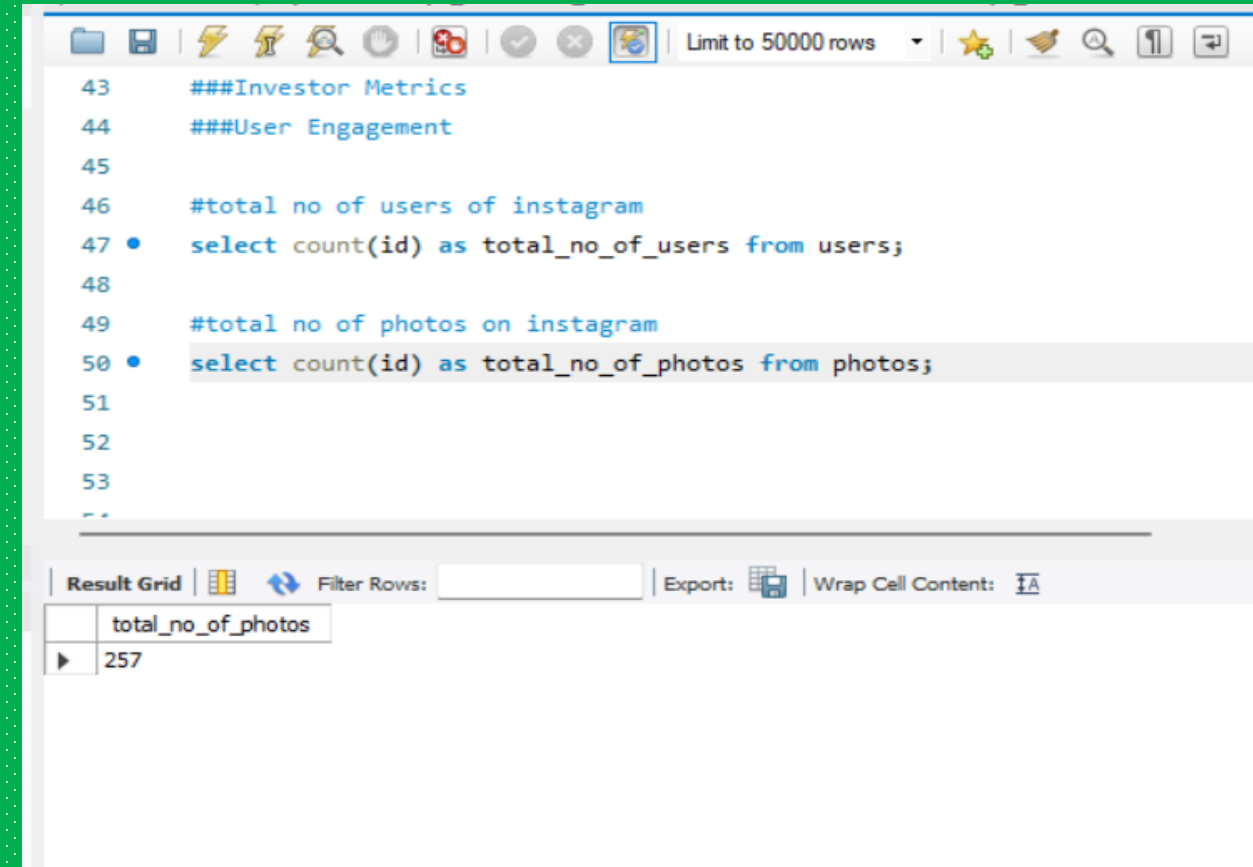
The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. A dropdown menu indicates 'Limit to 50000 rows'. The SQL editor contains the following code:

```
43  ###Investor Metrics
44  ###User Engagement
45
46  #total no of users of instagram
47  • select count(id) as total_no_of_users from users;
48
49
50
51
52
53
```

Below the editor, the 'Result Grid' tab is active, displaying a single row of results:

	total_no_of_users
▶	100

Total Number Of Users in Instagram



The screenshot shows the same SQL IDE interface. The SQL editor contains the following code:

```
43  ###Investor Metrics
44  ###User Engagement
45
46  #total no of users of instagram
47  • select count(id) as total_no_of_users from users;
48
49  #total no of photos on instagram
50  • select count(id) as total_no_of_photos from photos;
51
52
53
```

Below the editor, the 'Result Grid' tab is active, displaying a single row of results:

	total_no_of_photos
▶	257

Total Number of Photos in Instagram



```

51
52     #average post per user
53 •   with cte as
54     ( select users.id, count(photos.id) as post_per_user
55       from users
56       left join photos
57       on users.id = photos.user_id
58       group by users.id
59       order by post_per_user desc )
60     select avg(post_per_user) as "Average Post per User"
61     from cte;

```

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

	Average Post per User
▶	2.5700

Average Post per User

## 7. Bots & Fake Accounts :

```
63     ###Bots & Fake Accounts
64 •   with cte as
65     (select likes.user_id, count(likes.user_id) as no_of_likes
66      from photos
67      left join likes
68      on photos.id = likes.photo_id
69      left join users
70      on photos.user_id = users.id
71      group by likes.user_id)
72     select user_id, users.username, no_of_likes
73     from cte
74     left join users
75     on cte.user_id = users.id
76     where no_of_likes = 257;
77
```

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

	user_id	username	no_of_likes
▶	5	Aniya_Hackett	257
	14	Jadyn81	257
	21	Rocio33	257
	24	Maxwell.Halvorson	257
	36	Ollie_Ledner37	257
	41	Mckenna17	257
	54	Duane60	257
	57	Julien_Schmidt	257
	66	Mike.Auer39	257
	71	Nia_Haag	257
	75	Leslie67	257
	76	Janelle.Nikolaus81	257
	91	Bethany20	257

Result 55 x

These are the accounts that have liked all the photos that are available, Hence we can conclude that these are bots.

## **Result :**

Analysed many useful insights that could help the business to launch a new marketing campaign, decide on features to build for an app, track the success of the app by measuring user engagement and improve the experience altogether would help the business grow and this would help in making data driven decisions for the business.

It has expanded my knowledge in MySQL and also improved my querying language skills and this project has shown me where do I lag and this will help me to improve in concepts of join.