

PROJECT:

This project is about gathering all the insights from the dataset provided and do all the needed actions to get the desired output or result. This result can help the Product team to make required changes and informed decisions to cater their growth according to the insights.

DESCRIPTION:

The project is all about finding hidden patterns of users from the database and use it accordingly to make profound decisions that will eventually add up to the success of the product. The product team has provided a list of questions that they need answers to. My approach to all these questions will be by countering each question with a familiar query, combining some functions together, altering the queries a little bit to answer all of the questions. Things that I have found out throughout the project are proper usage of queries, implementing these queries on real-time database and real-time or real-world problems.

APPROACH:

My approach towards this set of questions was be in form of understanding them first, that is what the product team really wants to know, how much, and from where I can get it from the database. Then I started to build queries accordingly in my head and tried few as trial & error method. After a few attempts I finally figured out all the queries required to answer these questions. Some of the questions were tough and took some time but I did my best and invested my time into it and got the answers right. I had executed each and every query several times to make sure that the output I got was right for the question.

TECH-STACK USED:

The software I had used to make all of this happen is the MySQL WORKBENCH 8.0 version 8.0.31.

INSIGHTS:

Well the insights and knowledge that I gained while performing this project was phenomenal. I had never worked on such kind of SQL questions to test out my skills. I was also impressed by the end that I was able to complete the project because at the beginning of the project(Instagram User Analytics) it felt daunting to me. But after solving few queries I regained my confidence and solved the rest of the questions quickly. I got to know how these queries really work on real-world database, got an insight of how back-end developers might be programming to get all the insights their organization needs.

RESULT:

A) MARKETING QUESTIONS:

1. Find the 5 oldest users of the Instagram from the database provided.

```
91 • SELECT * from ig_clone.users;  
92 • select id, username, created_at from ig_clone.users order by created_at asc limit 5;  
93  
94  
95  
96
```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	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			
•	NULL	NULL	NULL			

Here I had used simple order by and ascending order sorting functions along with a limit of 5 as per the question which generates appropriate result required. If we want to find the oldest user then we may have to search from the beginning, to do this I had ordered the table in ascending order with a limit of 5.

2. Find the users who have never posted a single photo on Instagram.

```
96 • select * from ig_clone.photos;  
97 • select id, username from ig_clone.users where id not in (select user_id from ig_clone.photos);  
98
```

Result Grid			Filter Rows:	Edit:	Export/Import:
	id	username			
▶	5	Aniya_Hackett			
	7	Kassandra_Homenick			
	14	Jadyn81			
	21	Rocio33			
	24	Maxwell.Halvorson			
	25	Tierra.Trantow			
	34	Pearl7			
	36	Ollie_Ledner37			
	41	Mckenna17			
	45	David.Osinski47			
	49	Morgan.Kassulke			
	53	Linnea59			
	54	Duane60			
	57	Julien_Schmidt			
	66	Mike.Auer39			
	68	Franco_Keebler64			
	71	Nia_Haag			
	74	Hulda.Macejkovic			
	75	Leslie67			
	76	Janelle.Nikolaus81			
	80	Darby_Herzog			

80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20
•	NULL NULL

Here the task is to find out the users who have not posted anything on Instagram. To get the users I had used not in keyword which basically returns true if a particular condition is not present in the dataset. Photos table had all the id info which had posted something on Instagram, so it was easy to find out which users have not posted anything on Instagram by Not IN keyword.

3. Identify the winner of the contest and provide their details to the team.

```
153     select likes.photo_id , users.username , count(likes.user_id) as like_user
154     from ig_clone.likes likes
155     inner join ig_clone.photos photos
156         on likes.photo_id = photos.id
157     inner join ig_clone.users users
158         on photos.user_id = users.id
159     group by likes.photo_id, users.username
160     order by like_user desc
161     limit 1
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	photo_id	username	like_user		
▶	145	Zack_Kemmer93	48		

I would like to mention here that, for this question I have referred the video i.e. the solution because I had been stuck at this question for 2-3 days and could not come to any conclusion. I had finished every other questions before this one but this one was very tricky because the information was not clearly visible or cited in the database. So please evaluate accordingly, I have not watched the entire video for each question besides this one.

4. Identify and suggest the top 5 most commonly used hashtags on the platform

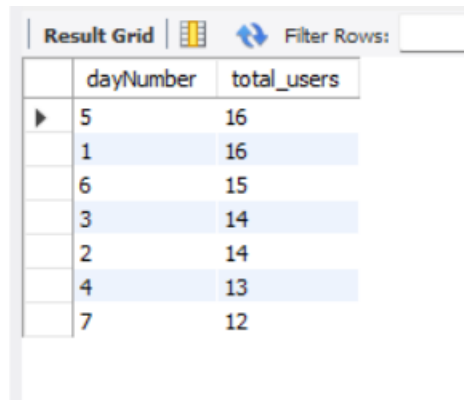
```
143 • select tag_id, count(tag_id) from ig_clone.photo_tags group by tag_id order by count(tag_id) desc limit 5;
```

Result Grid			Filter Rows:
	tag_id	count(tag_id)	
▶	21	59	
	20	42	
	17	39	
	13	38	
	18	24	

To find the 5 most commonly used hashtags we have to find count i.e. number of occurrence of a specific hashtag. To find this count I had accessed the photo_tags table then I have ordered the values by the count of each hashtag in descending order with a limit 5.

5. What day of the week do most users register on?

```
113 • select dayofweek(created_at) as dayNumber, count( id) as total_users from ig_clone.users group by dayofweek(created_at) order by total_users desc;
```



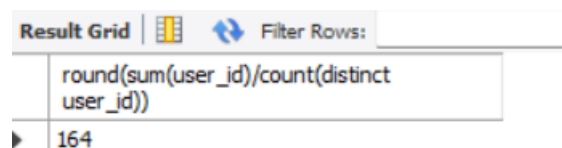
	dayNumber	total_users
▶	5	16
	1	16
	6	15
	3	14
	2	14
	4	13
	7	12

In MySQL or SQL there is a function called `dayofweek()` which converts the date format into an index number of a week. I had first converted the dates to these indexes then calculated their number of occurrence. Then I ordered this data on the count of each index occurrence in descending order. So according to the results, 5='Thursday' has most user registrations.

B) Investor Metrics:

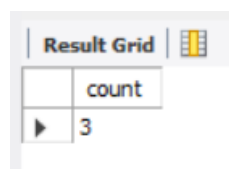
1. Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

```
133 • select round(sum(user_id)/count(distinct user_id)) from ig_clone.photos;
```



	round(sum(user_id)/count(distinct user_id))
▶	164

```
134 • select count(distinct image_url) from ig_clone.photos;  
135 • select round(count(distinct image_url)/count(distinct user_id)) as count from ig_clone.photos;  
136
```



	count
▶	3

I had rounded off the sum of user id by number of users to get the average post posted by a user on Instagram.

Also to provide an answer to the latter part of the question I have simply converted the question into a query to get the apt result.

2. Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

```
select user_id, count(user_id) from ig_clone.likes group by user_id order by count(user_id) desc ;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
user_id	count(user_id)		
21	257		
71	257		
5	257		
66	257		
41	257		
14	257		
57	257		
24	257		
76	257		
75	257		
54	257		
91	257		
36	257		
16	103		
96	98		
69	97		
65	96		
2	94		
--	--		

To find the number of bots, I have calculated the number of counts by each user. Since the total photos are 257, I have checked if the count of likes of any user is equal to 257. If any user has likes count equal to 257 then that user should be considered as a bot.