

INSTAGRAM PROJECT

Project Description: The overview of the project is to definitely to increase the company or app value and provide insights which will give or effect the growth of the Instagram. From the given data I would like to do whatever insights that I can draw from the data to improve the company growth. It can be useful to the for marketing team to analyze and to engage the users and it will be helpful to launch any advertisements in given days. To work on this data, I will use chose MYSQL.

Tech-Stack Used: Here I am using MYSQL workbench because it is free and user friendly. I have already had a work experience on this. Not only that but it is recommended also by trainity.



Approach:

#A) Marketing Analysis:

#1. Loyal User Reward: Identify the five oldest users on Instagram from the provided database. My moto here is to get the old users to give them the loyal user reward.

So, here I have planned to go by the users with their created date in descending order and used limit to restrict the users to 5 only.

```
select id, username, created_at from users order by created_at asc limit 5;
```

	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

#2. In active User Engagement: Identify users who have never posted a single photo on Instagram.

In this case my aim is to know the inactive users who never posted. So, here I thought to go by the user list or users that there is no one record in the photos table which give the users who never posted. Here I can use joins concept also but I think here the subquery concept gets covered.

`select b.id,b.username from users b where b.id not in (select distinct user_id from photos);`

	id	username
	5	Aniya_Hackett
	7	Kasandra_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	Lanelle.Nikolaus81

76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20
NULL	NULL

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

Here my aim to get the user who got more likes for a single photo. To get this I have joined users table and likes table and get count of the photo_id cause for every photo_id the likes will be get counted. To get the only one record I have used here limit and order by descending. Here I have option of using rank also. But I think it to be optimized query.

```
select b.username,a.photo_id,count(a.photo_id) likes from likes a,users b
where a.user_id=b.id group by photo_id order by count(a.photo_id) desc limit 1;
```

	username	photo_id	likes
▶	Harley_Lind18	145	48

#4. Hash tag Research: Identify and suggest the top five most commonly used hashtags on the platform.

Here my aim is to get the Hash tags which are used highly by so many people. To get this I have used the photo tags and tags tables and joined them by using tag_id and used count to know the value and joined to get the name of the tag used limit to get 5 records. Here we can use inner join also but I do like this in real time as oracle technical consultant. In the place of limit we can use ranking also.

```
select b.tag_name,count(a.tag_id) from photo_tags a, tags b
where a.tag_id=b.id group by a.tag_id,b.tag_name order by count(a.tag_id) desc limit 5;
```

	tag_name	count(a.tag_id)
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

#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.

Here in this case my aim is to get the Day of the weeks more number of registrations taken place. It can be very useful to launch any new Ads. To get this I have used the dayname function to get the name of the date from the users table. Here I have used the count of that week name and used dense_rank to rank the count based on the dayname.

```
select * from
(select dayname(created_at) Day_Name, count(dayname(created_at)) Registration_count,
dense_rank() over (order by count(dayname(created_at)) desc) ranking
from users
group by dayname(created_at)
order by count(dayname(created_at)) desc
) a
where a.ranking=1;
```

	Day_Name	Registration_count	ranking
▶	Thursday	16	1
	Sunday	16	1

#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.

In this case I have small misunderstanding so I have written 2 queries.

For the first query I have written like getting the average number of posts by the people who posted only. For more clarity I have used the posted users count and posts count also

```
select count(id) posts,count(distinct user_id) posted_users,count(id)/count(distinct user_id)
average_posts from photos;
```

	posts	posted_users	average_posts
▶	257	74	3.4730

For second one I have taken the total posts count and total users count and got the average of them.

```
select (select count(image_url) from photos a) total_posts,
(select count(id) from users) total_users,
(select count(image_url) from photos a)/(select count(id) from users)
Average_of_posts_based_on_total_users
from dual;
```

	total_posts	total_users	Average_of_posts_based_on_total_users
▶	257	100	2.5700

#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.

In this case to identify the total fake accounts the measure is to know all the accounts are users who liked all the photos. To get this I have first checked all the likes count and all the posts count also. So, here I have used the subquery to know the count of the photos and used the having clause to get the users who liked every single photo.

`select count(*) from likes;`

	count(*)
▶	8782

`select count(id) from photos;`

	count(id)
▶	257

`select b.username,a.user_id, count(a.user_id) total_likes from likes a,users b
where a.user_id=b.id
group by a.user_id having count(a.user_id) = (select count(id) from photos);`

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

Insights: While working this project I have gained the knowledge of questioning the data and working every pinpoint corner of data. Cause till now I didn't even know that based on the registration day also the ads will be released. While working on these issues I again recalled all my SQL knowledge and basics again which we do not work often. I feel good to participate in this, cause it improved my confidence than any other thing.

Results: I think in my two years of journey as a sql developer and data analyst I have never used dense_rank literally. But glad to witness that I am pretty confident after this project. Till now I know and had good experience in SQL but I thought for data analyst this would be enough.. But today I can say that I can do or write any type of SQL Query easily and confidently.

Please let me know if any mistakes are there and any query optimization can be done to improve accuracy and efficiency.

Thank you trainity.