

PROJECT 2

**INSTAGRAM
USER ANALYTICS**

A MADHAVA VARMA

DESCRIPTION

The project is to make an analysis of user interactions and engagement with the Instagram app. The primary purpose is to obtain insights that can be used to make strategic decisions for scaling up the business.

TECH-STACK USED

MySQL Workbench 8.0 CE and SQL are used for the project. They are used as MySQL is known for its performance, especially in cases where we handle large volumes of data, which is required for analyzing user interactions on a large platform like Instagram.

ANALYSIS ON

PART A- Marketing Analysis

- a. Loyal User Reward
- b. Inactive User Engagement
- c. Contest Winner Declaration
- d. Hashtag Research
- e. Ad Campaign Launch

PART B- Marketing Analysis

- a. User Engagement
- b. Bots & Fake Accounts

PART-A MARKETING ANALYSIS

(a) LOYAL USER AWARD

The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

Task: Identify the five oldest users on Instagram from the provided database.

STEPS:

1. We use the data provided in the **creaded_at** column from the **users** table to get the **users** who have been using the platform for the longest time.
2. We select the table **users** and use **ORDER BY** function for the **created_at** column to order the data in ascending order using **ASC**.
3. We can now use **limit** function to get the top 5 customers who are using the platform for the longest time.

PART-A MARKETING ANALYSIS

(a) LOYAL USER AWARD

QUERY:

```
select * from users  
order by created_at  
limit 5;
```

INSIGHT:

The following are the top 5 loyal customers on the Instagram platform.

RESULT

	<u>id</u>	<u>username</u>	<u>created_at</u>
▶	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
	HULL	HULL	HULL

PART-A MARKETING ANALYSIS

(b) Inactive User Engagement

The team wants to encourage inactive users to start posting by sending them promotional emails.

Task: Identify users who have never posted a single photo on Instagram.

STEPS:

1. We need to select **username** column from the **users** table.
2. Next, we need to use **left join** function to left join **photos** and **users** table **on users.id = photos.user_id** as both of them have common data.
3. Now we have to get the data where **photos.id** is **NULL** from the **users** table.

PART-A MARKETING ANALYSIS

(b) Inactive User Engagement

QUERY:

```
select username, users.id  
from users  
left join photos  
on users.id = photos.user_id  
where photos.id IS NULL;
```

INSIGHT:

There are a total of 26 users out of the 100 users who have never posted a single photo.

RESULT:

	username	id
▶	Aniya_Hackett	5
	Kasandra_Homenick	7
	Jadyn81	14
	Rocio33	21
	Maxwell.Halvorson	24
	Tierra.Trantow	25
	Pearl7	34
	Ollie_Ledner37	36
	Mckenna17	41
	David.Osinski47	45
	Morgan.Kassulke	49
	Linnea59	53
	Duane60	54
	Julien_Schmidt	57
	Mike.Auer39	66
	Franco_Keebler64	68
	Nia_Haag	71
	Hulda.Macejkovic	74
	Leslie67	75
	Janelle.Nikolaus81	76
	Darby_Herzog	80
	Esther.Zulauf61	81
	Bartholome.Bernhard	83
	Jessyca_West	89
	Esmeralda.Mraz57	90
	Bethany20	91

PART-A MARKETING ANALYSIS

(c) Contest Winner Declaration:

The team has organized a contest where the user with the most likes on a single photo wins.

Task: Determine the winner of the contest and provide their details to the team.

STEPS:

1. We need to select **users.username**, **photos.id**, **photos.image_url** and **count(*)** as **total_likes** .
2. Next, we need to use **inner join** function to inner join **users**, **photos** and **likes** table **on likes.photo_id = photos.id** and **photos.user_id = users.id**.
3. Now we need to use **GROUP BY** function to group the data based on **photos.id**.
4. Then, we use **ORDER BY** function to sort the data based on **total_likes** in descending order using **DESC** and limit it to get the winner by using the **limit function**.

PART-A MARKETING ANALYSIS

(c) Contest Winner Declaration

QUERY:

```
select users.id as user_id , users.username,  
photos.id as p_id, photos.image_url,  
count(*) as total_likes
```

```
from photos inner join likes  
on likes.photo_id = photos.id  
inner join users  
on photos.user_id = users.id  
  
group by photos.id  
order by total_likes DESC  
limit 1;
```

RESULT:

	user_id	username	p_id	image_url	total_likes
▶	52	Zack_Kemmer93	145	https://jarret.name	48

INSIGHT:

Hence, the user with **username Zack_Kemmer93** with **user_id 52** is the winner of the contest for having the most **number of likes (48)** for his photo with **photo_id 145**.

PART-A MARKETING ANALYSIS

(d) Hashtag Research:

A partner brand wants to know the most popular hashtags to use in their posts to reach the most people.

Task: Identify and suggest the top five most commonly used hashtags on the platform.

STEPS:

1. Select **tags.tag_name, count(*) as tag_count** to count the number of tags individually.
2. Now we need to use **JOIN** function to join **tags** and **photo_tags** tables on **tags.id = photo_tags.id** as they contain common data.
3. Next, we use **GROUP BY** function to group the data based on **tags.tag_name**.
4. Then, we use **ORDER BY** function to sort the data based on **tag_count** in descending order using **DESC** and limit it to get the winner by using the **limit function**.

PART-A MARKETING ANALYSIS

(d) Hashtag Research:

QUERY:

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

INSIGHT:

Hence, the following are the most used hashtags on the Instagram platform.

RESULT:

	tag_name	tag_count
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

PART-A MARKETING ANALYSIS

(e) Launching AD Campaign:

The team wants to know the best day of the week to launch ads.

Task: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

STEPS:

1. Select **dayname(created_at)** as **day_in_week**, **count(*)** as **user_count** from the **users** table.
2. Now, we use **GROUP BY** function to group the data based on **day_in_week**.
3. Then, we use **ORDER BY** function to sort the data based on **user_count** in descending order using **DESC**.

PART-A MARKETING ANALYSIS

(e) Launching AD Campaign:

QUERY:

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

INSIGHT:

Most of the users registered on Thursday and Sunday. Hence it would be beneficial to start the AD Campaigns on Thursdays and Saturdays.

RESULT:

	day_in_week	user_count
▶	Thursday	16
	Sunday	16
	Friday	15
	Tuesday	14
	Monday	14
	Wednesday	13
	Saturday	12

PART-B INVESTOR METRICS

(a) USER ENGAGEMENT

Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.

Task: 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.

STEPS:

1. First, we need to get the count of posts/photos present in **photos.id** column of **photos** table using **count(*) from photos**.
2. Next, we need to get the count of users present in the **users.id** column of **users** table using **count(*) from users**.
3. Now, we need to divide the **total number of posts / total number of users** to get the **average** number of posts per user.

PART-B INVESTOR METRICS

(a) USER ENGAGEMENT

QUERY:

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

RESULT:

	avg_posts_per_user
▶	2.5700

INSIGHT:

The **average number of posts per user** on the platform is found to be **2.57**.

There are **257 photos** in **photos** table and there are **100 users** in the **users** table and the division of the **total number of photos / total number of users** is **257/100** which is equal to **2.57**.

PART-B INVESTOR METRICS

(b) Bots and Fake Accounts

Investors want to know if the platform is crowded with fake and dummy accounts.

Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

STEPS:

1. First, we need to select the **user_id** column in **photos** table, **username** column from **users** table and **count(*) as total_likes** to count the total number of likes.
2. Next, we need to use the **inner join** function to inner join **users** and **likes** tables based **on users.id = likes.user_id**.
3. Now, we need to use **group by** function to get the data based on **likes.user_id**.
4. Finally, we need to get the data of users from **photos** table who are **having count(*) equal to the value of total_likes**.

PART-B INVESTOR METRICS

(b) Bots and Fake Accounts

QUERY:

```
select user_id, username, count(*) as total_likes  
from users  
inner join likes  
on users.id = likes.user_id  
group by likes.user_id  
having (select count(*) from photos) = total_likes;
```

INSIGHT:

Hence, we can conclude that the following 13 users are potentially bots and the corresponding accounts are potentially fake or dummy accounts.

RESULT:

	user_id	username	total_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

Working on this project helped me understand the concepts of SQL from basics to advanced level. It helped me gain experience on handling MySql effectively and how Data Analytics is implemented using it in the real world applications such as the platform of Instagram.

**THANK
YOU.**

The background features a minimalist design with black wavy lines on a white surface. A large, bold, black sans-serif font displays the words "THANK" and "YOU." stacked vertically in the center. The left side of the image shows a bundle of wavy lines converging towards the center, while the right side shows a series of parallel, flowing wavy lines extending from the center towards the edge.