

PROJECT – 2

Instagram User Analytics



Name : Deeksha S Rella



TABLE OF CONTENTS

- Project Description
- Approach
- Tech-Stack Used
- Insights
- Result



PROJECT DESCRIPTION

A user analysis of Instagram is necessary for the project, which includes information on the most devoted users, inactive users, most-liked photos, hashtag research, and the launch of an advertising campaign.

For the investors, insights are also needed on user engagement, bots, and fake accounts.

Mainly, locating business insights that may be applied by teams throughout the company to start a new marketing campaign decide which features to include in an app, monitor the app's success by gauging user engagement, and enhance the overall experience while promoting corporate growth.



APPROACH

Database creation: Using the DDL and DML SQL queries provided by the product manager (as per project) in the MySQL database, the values were created and entered using MySQL Workbench.

Extraction of insights: Following database creation, necessary insights are derived from the database tables by executing SQL queries in MySQL Workbench.

TECH-STACK USED

I utilized SQL (Ver 8.0.34), which I downloaded from the document's URL, and then I downloaded SQL WORKBENCH (Ver 8.0.34), which offers a visual console for managing MySQL systems and improving database visibility.



INSIGHTS

A. Marketing

1. Rewarding the Most Consistent Users : Individuals that have used the platform the longest.

USE ig_clone;
select * from users
order by date(created_at)
limit 5

```
91
92 * USE ig_clone;
93 * select * from users
94   order by date(created_at)
95   limit 5
96
```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Contents:	Fetch rows:
	id	username	created_at				
▶	80	Darby_Herzog	2016-05-06 00:14:21				
	67	Emiko_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				
*	id	username	created_at				



2. Inactive User Engagement to reactivate : By sending them promotional emails to upload their initial image.

```
select users.username, users.id  
from users
```

```
where not exists (select* from photos where users.id = photos.user_id)  
order by username
```

```
97  
98 • select users.username, users.id  
99   from users  
100  where not exists (select* from photos where users.id = photos.user_id)  
101  order by username
```

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

	username	id
▶	Aniya_Hackett	5
	Bartholome.Bernhard	83
	Bethany20	91
	Darby_Herzog	80
	David.Osinski47	45
	Duane60	54
	Esmeralda.Mraz57	90
	Esther.Zulauf61	81
	Franco_Keebler64	68
	Hulda.Macejkovic	74
	Jadyn81	14
	Janelle.Nikolaus81	76
	Jessyca_West	89
	Julien_Schmidt	57
	Kassandra_Homenick	7

users 3 x



3. Declaring the Contest Winner : The participant who receives the most likes on a single photo wins the contest that the team has set up.

Use ig_clone;

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

```
from likes
```

```
join photos on photos.id=likes.photo_id
```

```
join users on users.id=likes.photo_id
```

```
group by photos.id
```

```
order by total_likes desc
```

```
limit 1;
```

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
112 • Use ig_clone;
113 • select users.username, photos.id, photos.image_url, count(*) as total_likes
114 • from likes
115 • join photos on photos.id=likes.photo_id
116 • join users on users.id=likes.photo_id
117 • group by photos.id
118 • order by total_likes desc
119 • limit 1;
```

The result grid shows the following data:

username	id	image_url	total_likes
Kaley9	30	http://kenny.com	41



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

```
select tags.tag_name,  
count(photo_tags.photo_id) as num_tags  
from photo_tags  
inner join tags on photo_tags.tag_id=tags.id  
group by tag_name  
order by num_tags desc  
limit 5
```

The screenshot shows a SQL query editor window titled "Query 1" with a tab "SQL File 3". The query is as follows:

```
select tags.tag_name, count(photo_tags.photo_id) as num_tags  
from photo_tags  
inner join tags on photo_tags.tag_id=tags.id  
group by tag_name  
order by num_tags desc  
limit 5
```

Below the query, the "Result Grid" shows the following data:

tag_name	num_tags
smile	59
beach	42
party	39
fun	38
concert	24

The interface includes various toolbars for editing, running queries, and viewing results. The status bar at the bottom indicates "Result 7 x".



5. Launch Ad Campaign : The team wants to know the best day of the week to launch ads.

```
select dayname(created_at) as day, count(*) as total
from users
group by day
order by total desc
```

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
136
137 • select dayname(created_at) as day, count(*) as total
138     from users
139     group by day
140     order by total desc
141
142
143
```

The result grid displays the following data:

day	total
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12



B. Investor Metrics

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

with cte as (

select users.id as userid, count(photos.id) as photoid

from users

left join photos on users.id=photos.user_id

group by (userid))

select sum(photoid) as total_photos, count(userid) as total_users, sum(photoid)/count(userid) as

photos_per_person

from cte

The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```
143 with cte as (  
144     select users.id as userid, count(photos.id) as photoid  
145     from users  
146     left join photos on users.id=photos.user_id  
147     group by (userid))  
148 select sum(photoid) as total_photos, count(userid) as total_users, sum(photoid)/count(userid) as photos_per_person  
149 from cte  
150  
151
```

The results grid shows the following data:

	total_photos	total_users	photos_per_person
▶	257	100	2.5700



2. Bots & Fake Accounts: Investors want to know if the platform is crowded with fake and dummy accounts.

```
select id, username
from users
where id in (select user_id
from likes
group by user_id
having count(user_id) = (select count(id)
from photos));
```

```
172
173 • select id,
174     username
175     from users
176     where id in (select user_id
177                 from likes
178                 group by user_id
179                 having count(user_id) = (select count(id)
180                                         from photos));
181
182
183
```

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

	id	username
▶	5	Aniya_Hackett
	14	Jadyn81
	21	Rocio33
	24	Maxwell.Halvorson
	36	Ollie_Ledner37
	41	Mckenna17
	54	Duane60
	57	Julien_Schmidt
	66	Mike_Auer39
	71	Nia_Haag
	75	Leslie67

users 1 x



RESULT

Working on this project has given me a foundational understanding of SQL queries and my query workbench. This project helped me understand the theory behind how queries are written and run, hinter a question. It has aided me by establishing the groundwork for my journey of learning SQL.