



# INSTAGRAM USER ANALYTICS

*By:- Vishal Singh Sangral*



# PROJECT DESCRIPTION

Purpose: Analyze Instagram user data to inform decision-making and future development.

Approach:

---

- Data Exploration: Understand the database schema, including users, posts, likes, and comments.
- Exploratory Analysis: Calculate overall user engagement metrics.
- Engagement Metrics: Compute average likes and comments per post.
- Top Users: Identify top 5 users with the highest followers.
- Time Analysis: Determine peak user activity times.
- Post Type Impact: Analyze the impact of different post types on engagement.
- Refinement: Iteratively refine queries based on findings and feedback.
- Documentation: Document queries, assumptions, and create a presentation for the product team.

Goal: Derive actionable insights to influence the future direction and growth of the Instagram app.



# SUMMARY FOR ALL THE KEY POINTS COVERED IN THIS PROJECT

**Project Purpose:** Analyze user interactions on Instagram to inform strategic decisions

**Teams Involved:** Marketing, Product, Development.

**Tools:** SQL, MySQL Workbench.

**Goals:** Identify loyal users, engage inactive users, declare contest winner, suggest popular hashtags, determine optimal ad launch day.

**Approach:** Leverage SQL queries to extract insights on user activity, post engagement, and platform dynamics.

## Marketing Analysis:

- Identify oldest users for loyalty rewards.
- Find inactive users for targeted promotional emails.
- Determine contest winner with most likes.
- Suggest top five hashtags for partner brand.
- Provide insights on optimal ad campaign launch day.

## Investor Metrics:

- Calculate average posts per user for engagement assessment.
- Identify potential bots by detecting users liking every photo.

**Insights:** Drive informed decisions for growth, user experience, and content strategy.

# Market Analysis

# LOYAL REWARD CUSTOMER:

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, database management, and execution. The left sidebar shows the 'SCHEMAS' tree with 'ig\_clone' selected, containing tables like 'comments', 'follows', 'likes', 'photos', 'photo\_tags', and 'tags'. The main editor displays a SQL query:

```
1 • USE ig_clone;
2
3 -- 1 Loyal User Reward: The marketing team wants to reward the most loyal users,
4 -- i.e., those who have been using the platform for the longest time.
5 -- My Task: Identify the five oldest users on Instagram from the provided database.
6
7 • Select * from users
8   order by created_at
9   limit 5;
```

Below the query editor, the 'Result Grid' is visible, 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
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Session'.

These 5 customers will get rewarded for being the oldest user of INSTAGRAM

# INACTIVE USER ENGAGEMENTS:

SCHEMAS

Filter objects

ig\_clone

Tables

comments

follows

likes

photos

photo\_tags

tags

Object Info

Session

No object selected

Limit to 1000 rows

10

11 -- 2 Inactive User Engagement: The team wants to encourage inactive users to start posting by sending them promotional emails.

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

13

14 • select \* from users u

15 left join photos p on u.id = p.user\_id

16 where p.id is null;

17

100%

20:16

Result Grid

Filter Rows: Search

Export:

	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	Jaclyn81	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
	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

Result 26

Read Only

Action Output

Time

Action

Response

Duration / Fetch Time

35

16:15:28

select \* from users u left join photos p on u.id = p.user\_id where p.id is null LIMIT 0, 1000

26 row(s) returned

0.0089 sec / 0.00001...

26 USERS ARE INACTIVE AND NOT POSTING OR NOT DOING ANY ACTIVITY ON INSTAGRAM

# CONTEST WINNER DECLARATION:

The screenshot shows a database management interface with a sidebar on the left containing a tree view of schemas and tables. The main area displays a SQL query and its results. The query is a complex join statement that finds the user with the most likes on a single photo. The results are shown in a table with columns for photo\_id, username, and like\_count. The result shows that user Zack\_Kemmer93 has 48 likes on photo 145.

Local instance 3306 - Warning - not supported

Administration Schemas PROJECT2

SCHEMAS

Filter objects

ig\_clone

Tables

- comments
- follows
- likes
- photos
- photo\_tags
- tags

Object Info Session

No object selected

Limit to 1000 rows

3 Contest Winner Declaration: The team has organized a contest where the user with the most likes on a single photo wins.  
My Task: Determine the winner of the contest and provide their details to the team.

```
20
21 • select count(*) from likes;
22 • select * from likes;
23 • select count(*) from users;
24 • select * from users;
25 • select * from photos;
26 • select count(*) from photos;
27
28 • select l.photo_id,u.username, count(l.user_id) like_count
29   from likes l
30  inner join photos p on l.photo_id = p.id
31  inner join users u on p.user_id = u.id
32  group by l.photo_id , u.username
33  order by like_count desc
34  limit 1;
```

100% 9:34

Result Grid

photo_id	username	like_count
145	Zack_Kemmer93	48

Result 28

Action Output

	Time	Action	Response	Duration / Fetch Time
36	17:07:20	select l.photo_id,u.username, count(l.user_id) like_count from likes l inner join photos p on l.photo_id = p.id inner join...	1 row(s) returned	0.023 sec / 0.000014...
37	17:07:50	select l.photo_id,u.username, count(l.user_id) like_count from likes l inner join photos p on l.photo_id = p.id inner join...	1 row(s) returned	0.018 sec / 0.000012...

ZACK\_KEMMER93 IS THE CONTEST WINNER FOR  
PHOTO\_ID 145 WITH 48 LIKES.

# HASHTAG RESEARCH:

SCHEMAS

Filter objects

ig\_clone

Tables

comments

follows

likes

photos

photo\_tags

tags

Object Info

Session

No object selected

Limit to 1000 rows

37

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

38

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

39

• select \* from photo\_tags;

40

• select \* from tags;

41

42

43

• select count(photo\_id) noofphotos, tag\_id , tag\_name

44

from photo\_tags p

45

join tags t

46

on p.tag\_id = t.id

47

group by tag\_id

48

order by noofphotos desc

49

limit 5;

50

100%

9:49

Result Grid

Filter Rows:

Search

Export:

Fetch rows:

	noofphotos	tag_id	tag_name
59	21	smile	
42	20	beach	
39	17	party	
38	13	fun	
24	18	concert	

Result 29

Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
37	17:07:50	select l.photo_id,u.username, count(l.user_id) like_count from likes l inner join photos p on l.photo_id = p.id inner join...	1 row(s) returned	0.018 sec / 0.000012...
38	17:16:00	select count(photo_id) noofphotos, tag_id , tag_name from photo_tags p join tags t on p.tag_id = t.id group by tag_id...	5 row(s) returned	0.060 sec / 0.000012...

TOP 5 HASHTAGS ARE: SMILE , BEACH , PARTY, FUN, CONCERT



# AD CAMPAIGN LAUNCH:

Administration

Schemas

PROJECT2

SCHEMAS

Filter objects

ig\_clone

Tables

comments

follows

likes

photos

photo\_tags

tags

Object Info

Session

No object selected

Limit to 1000 rows

50

51 -- 5 Ad Campaign Launch: The team wants to know the best day of the week to launch ads.

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

53

54 • select \* from users;

55 • select date\_format(created\_at, '%W') day ,count(username) num\_of\_user

56 from users

57 group by day

58 order by num\_of\_user desc;

59

100%

27:58

Result Grid

Filter Rows: Search

Export

day	num_of_user
4	16
0	16
5	15
2	14
1	14
3	13
6	12

Result 30

Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 32	12:57:27	USE ig_clone	0 row(s) affected	0.0024 sec
✓ 33	12:58:22	Select * from users order by created_at limit 5	5 row(s) returned	0.0032 sec / 0.00002...
✓ 34	12:59:04	Select * from users order by created_at limit 5	5 row(s) returned	0.00090 sec / 0.000...

Query Completed

0 AND 4 THAT IS SUNDAY AND WEDNESDAY ARE TWO DAYS WHEN  
USERS SIGN UP ON INSTAGRAM MOST ,  
SO THESE TWO COULD BE TWO BEST DAYS TO LAUNCH THE AD  
CAMPAIGN

# INVESTOR METRICS

# USER ENGAGEMENT:

The screenshot shows a database management interface with a sidebar on the left containing a tree view of schemas. The main area displays a SQL query and its results. The query is as follows:

```
61 -- 6 User Engagement: Investors want to know if users are still active and posting on Instagram or if they are making fewer posts.
62 -- Calculate the average number of posts per user on Instagram.
63 -- Also, provide the total number of photos on Instagram divided by the total number of users''
64 • select * from photos;
65 • select * from users;
66
67 • with base as
68   (select count(p.id) as num_of_photos , u.id as user_id
69    from users u
70    left join photos p
71     on u.id = p.user_id
72    group by u.id)
73
74   select sum(num_of_photos) as total_photos,
75          count(user_id) as total_users,
76          sum(num_of_photos)/count(user_id) as avg_photo_per_user
77   from base;
78
```

The results are displayed in a table with the following data:

total_phot...	total_users	avg_photo_per_u...
257	100	2.5700

Below the results, there is an 'Action Output' section showing a log of database actions and their responses.

	Time	Action	Response	Duration / Fetch Time
✓ 35	16:15:28	select * from users u left join photos p on u.id = p.user_id where p.id is null LIMIT 0, 1000	26 row(s) returned	0.0089 sec / 0.00001...
✓ 36	17:07:20	select l.photo_id,u.username, count(l.user_id) like_count from likes l inner join photos p on l.photo_id = p.id inner join...	1 row(s) returned	0.023 sec / 0.000014...
✓ 37	17:07:50	select l.photo_id,u.username, count(l.user_id) like_count from likes l inner join photos p on l.photo_id = p.id inner join...	1 row(s) returned	0.018 sec / 0.000012...
✓ 38	17:16:00	select count(photo_id) noofphotos, tag_id , tag_name from photo_tags p join tags t on p.tag_id = t.id group by tag_id...	5 row(s) returned	0.060 sec / 0.000012...
✓ 39	17:20:05	select date_format((created_at),'%w') day ,count(username) num_of_user from users group by day order by num_of ...	7 row(s) returned	0.0018 sec / 0.00000...

THERE ARE 100 USERS AND TOTAL 257 PHOTOS WHICH GIVING US 2.57 AVERAGE PHOTO PER USERS ON INSTAGRAM

# BOT & FAKE ACCOUNTS

Administration

Schemas

PROJECT2

SCHEMAS

Filter objects

ig\_clone

Tables

comments

follows

likes

photos

photo\_tags

tags

Object Info

Session

No object selected

Limit to 1000 rows

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

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

79

80

81

82 • Select \* from likes;

83 • select \* from users;

84 • with cte as

85 (select user\_id,count(photo\_id) photo\_liked

86 from likes

87 group by user\_id)

88

89 select user\_id,username, photo\_liked

90 from cte

91 inner join users

92 on user\_id = id

93 where photo\_liked = (select count(\*) from photos);

94

75%

51:93

Result Grid

Filter Rows:

Search

Export:

user\_id

username

photo\_liked

5

Aniya\_Hackett

257

14

Jaclyn81

257

21

Rocio33

257

24

Maxwell.Halvorson

257

36

Ollie\_Ledner37

257

41

Mckenna17

257

54

D......

257

Result 32

Read Only

Action Output

Time

Action

Response

Duration / Fetch Time

37

17:16:00

select count(photo\_id) noofphotos, tag\_id , tag\_name from photo\_tags p join tags t on p.tag\_id = t.id group by tag\_id...

5 row(s) returned

0.060 sec / 0.000012...

38

17:16:00

select count(photo\_id) noofphotos, tag\_id , tag\_name from photo\_tags p join tags t on p.tag\_id = t.id group by tag\_id...

5 row(s) returned

0.060 sec / 0.000012...

39

17:20:05

select date\_format((created\_at),'%w') day ,count(username) num\_of\_user from users group by day order by num\_of\_...

7 row(s) returned

0.0018 sec / 0.00000...

40

17:26:38

with base as (select count(p.id) as num\_of\_photos , u.id as user\_id from users u left join photos p on u.id = p.user\_id...

1 row(s) returned

0.0018 sec / 0.00001...

41

17:28:17

with cte as (select user\_id,count(photo\_id) photo\_liked from likes group by user\_id) select user\_id,username, photo\_...

13 row(s) returned

0.0073 sec / 0.00001...

THERE ARE 13 ACCOUNTS WHO LIKED EVERY SINGLE PHOTO ON INSTAGRAM

EXTRA WORK

Local instance 3306 - Warning - not supported

Administration

Schemas

PROJECT2

SCHEMAS

Filter objects

ig\_clone

Tables

comments

follows

likes

photos

photo\_tags

tags

Object Info

Session

No object selected

Limit to 1000 rows

```
94
95  -- EXTRA WORKS (number of photos posted by each users)
96
97  • select * from photos;
98
99  • select user_id, count(id) photos_posted
100    from photos
101    group by user_id
102    order by photos_posted desc;
103
104  -- user id 23 is the most engaged user on instagram he posted 12 photos his user name is EVELINE
105  • select * from users
106    where id = 23;
107  -- only 74 users are active on instagram out of 100 users.
108  -- Resulting 26 users out of which 13 are fake bot accounts and 13 are inactive users .
109
```

75%

40:99

Result Grid

Filter Rows:

Search

Export:

	user_id	photos_posted
59	10	
86	9	
58	8	
29	8	
77	6	
1	5	

Result 41

Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
48	17:40:02	select user_id, count(id) photos_posted from photos group by user_id order by photos_posted desc LIMIT 0, 1000	74 row(s) returned	0.00072 sec / 0.0000...
49	17:41:58	select * from users LIMIT 0, 1000	100 row(s) returned	0.0011 sec / 0.00006...
50	17:42:11	select * from users where id = 23 LIMIT 0, 1000	1 row(s) returned	0.00064 sec / 0.000...
51	17:43:38	select user_id, count(id) photos_posted from photos group by user_id order by photos_posted desc LIMIT 0, 1000	74 row(s) returned	0.00072 sec / 0.0000...

Thank You Trainity