

# INSTAGRAM USER ANALYTICS

---

**Sowmiya R** 

## PROJECT DESCRIPTION

The aim is to provide a insights on the questions asked by the management team by finding marketing metrics and investor metrics in the instagram cloning database.

## APPROACH

First of all , I had compiled a provided instagram dataset into a query compiler . Then I had understand each and every requirements with the provided requirements.

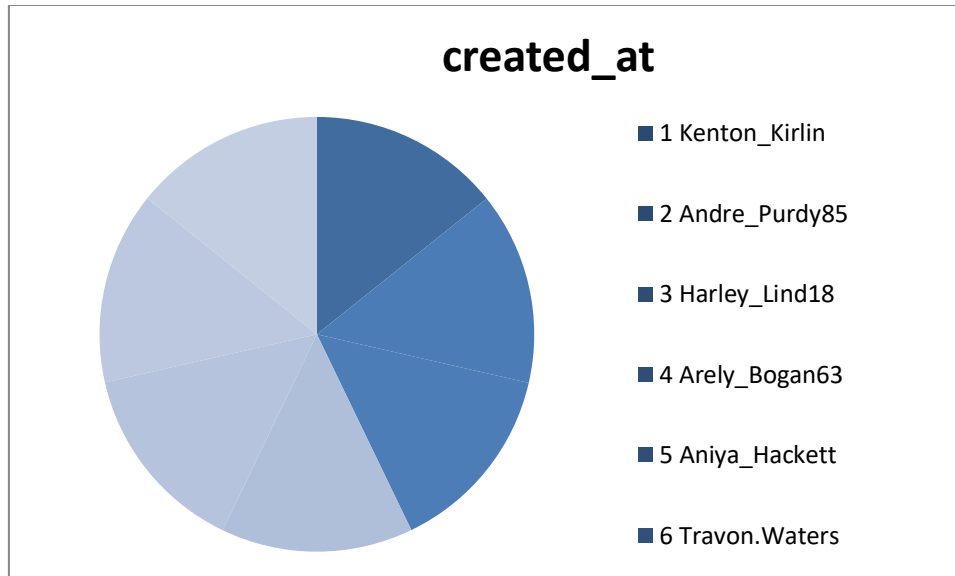
I had used a online query compiler to provide a valuable insights to the possible questions they have asked below in this analysis.

## TECH STACK

I have used MySQL 8.0.33 and MySQL is a widely used open-source relational database management system (RDBMS) that offers several benefits when it comes to data analytics. Meanwhile can analyze to provide a valuable insights in the instagram dataset.

## INSIGHTS

Let's answering the each possible questions of each metrics with the mysql code as follows:



A) **MARKETING METRICS** : The marketing team wants to launch some campaigns, and they need your help with the following:

**1.Rewarding Most Loyal Users:** Finding the 5 oldest users of the Instagram from the database provided with the joined date.

```
SELECT *FROM users
ORDER BY created_at DESC LIMIT 5;
```

```

48 CREATE TABLE tags(
49   id INTEGER AUTO-INCREMENT PRIMARY KEY,
50   tag_name VARCHAR(255) UNIQUE NOT NULL,
51   created_at TIMESTAMP DEFAULT NOW()
52 );
53
54 --projection table: Photos - Tags%
55
56 CREATE TABLE photo_tags(
57   photo_id INT NOT NULL,
58   tag_id INT NOT NULL,
59   FOREIGN KEY(photo_id) REFERENCES photos(id),
60   FOREIGN KEY(tag_id) REFERENCES tags(id),
61   PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65 INSERT INTO users(username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.880'), ('Andre_Purdy85', '2017-04-0
66
67
68 INSERT INTO photos(image_url, user_id) VALUES ('http://elishah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1)
69
70
71 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
72
73
74 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolorem', 2, 1), ('que ex duimus', 3, 1), ('alias
75
76
77 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (5, 1), (10, 1), (11, 1), (14, 1), (15, 1), (21, 1), (24, 1), (
78
79
80 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
81
82
83 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 10), (1, 17), (1, 23), (1, 33), (1, 39), (2, 4), (2, 3), (2, 20), (2,
84
85
86
87
88
89
90 SELECT *FROM users ORDER BY created_at DESC LIMIT 5;
91

```

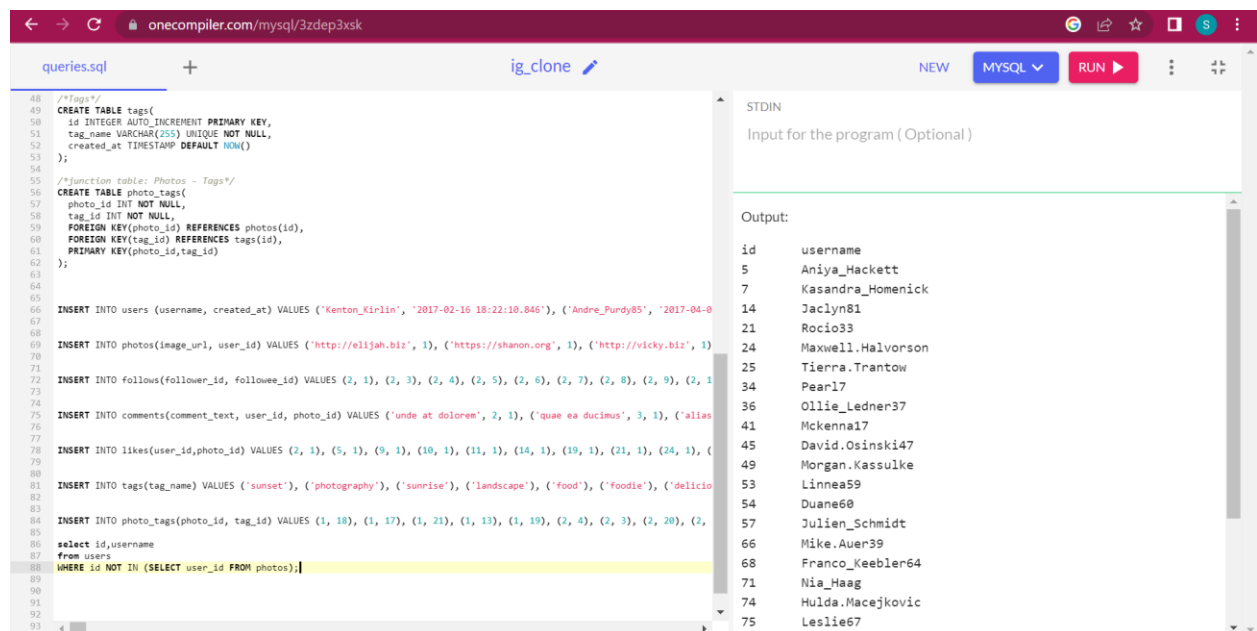
id	username	created_at
11	Justina.Gaylord27	2017-05-04 16:32:16
6	Travon.Waters	2017-04-30 13:26:14
85	Milford_Gleichner42	2017-04-30 07:50:51
19	Hailee26	2017-04-29 18:53:40
24	Maxwell.Halvorson	2017-04-18 02:32:44

Oldest 5 users joining date with their username and user id:

id	username	created_at
38	Jordyn.Jacobson2	2016-05-14 07:56:26
63	Elenor88	2016-05-08 01:30:41
67	Emilio_Bernier52	2016-05-06 13:04:30
80	Darby_Herzog	2016-05-06 00:14:21
95	Nicole71	2016-05-06 07:30:22

**2.Remind Inactive Users to Start Posting:** By sending them promotional emails to post their 1st photo. Find the users who have never posted a single photo on Instagram.

```
select id,username
from users
WHERE id NOT IN (SELECT user_id FROM photos);
```



The screenshot shows a web-based MySQL IDE interface. The left pane displays a SQL script with various database operations including table creation, foreign key setup, and data insertion. The right pane shows the output of a query, which is a list of user IDs and usernames. The query being executed is highlighted in the left pane.

```
48 /*tags*/
49 CREATE TABLE tags(
50   id INTEGER AUTO_INCREMENT PRIMARY KEY,
51   tag_name VARCHAR(255) UNIQUE NOT NULL,
52   created_at TIMESTAMP DEFAULT NOW()
53 );
54
55 /*junction table: Photos - Tags*/
56 CREATE TABLE photo_tags(
57   photo_id INT NOT NULL,
58   tag_id INT NOT NULL,
59   FOREIGN KEY(photo_id) REFERENCES photos(id),
60   FOREIGN KEY(tag_id) REFERENCES tags(id),
61   PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65
66 INSERT INTO users(username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-0
67
68 INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shamon.org', 1), ('http://vicky.biz', 1)
69
70 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
71
72 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolore', 2, 1), ('quae ea ducimus', 3, 1), ('alias
73
74
75 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
76
77
78 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
79
80
81 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
82
83
84
85
86 select id,username
87 from users
88 WHERE id NOT IN (SELECT user_id FROM photos);
89
90
91
92
93
```

Output:

id	username
5	Aniya_Hackett
7	Kassandra_Homenick
14	Jaclyn81
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

There are total 26 users who never posted a photo on the platform:

id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
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
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
<b>91</b>	Bethany20

**3.Declaring Contest Winner:** The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner. Identify the winner of the contest and provide their details to the team.

```
SELECT username, photos.id, photos.image_url, count(*) as total 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 DESC LIMIT 1;
```

The screenshot shows a MySQL query editor with a schema for an Instagram-like application. The schema includes tables for users, photos, tags, photo\_tags, follows, comments, and likes. A query is executed to find the user with the most likes on a single photo. The output shows that the user 'Zack\_Kemmer93' has 48 likes for a single photo.

```
queries.sql + ig_clone NEW MySQL RUN
```

```
47 --
48 /*Tags*/
49 CREATE TABLE tags(
50   id INTEGER AUTO INCREMENT PRIMARY KEY,
51   tag_name VARCHAR(255) UNIQUE NOT NULL,
52   created_at TIMESTAMP DEFAULT NOW()
53 );
54
55 /*function tables: Photos - Tags*/
56 CREATE TABLE photo_tags(
57   photo_id INT NOT NULL,
58   tag_id INT NOT NULL,
59   FOREIGN KEY(photo_id) REFERENCES photos(id),
60   FOREIGN KEY(tag_id) REFERENCES tags(id),
61   PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65
66 INSERT INTO users (username, created_at) VALUES ('Kenton_Kirilin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-0
67
68
69 INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1)
70
71
72 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
73
74
75 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolore', 2, 1), ('quae ea ducimus', 3, 1), ('alias
76
77
78 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
79
80
81 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
82
83
84 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
85
86
87 select username,photos.id,photos.image_url,count(*) as total from photos
88 inner join likes on likes.photo_id = photos.id
89 inner join users on photos.user_id = users.id group by photos.id order by total DESC limit 1;
```

STDIN  
Input for the program ( Optional )

Output:

username	id	image_url	total
Zack_Kemmer93	145	https://jarret.name	48

User with ID: Zack\_kemmer93 has won the contest with 48 likes for a single photo he had posted.

username	id	image_url	total
Zack_Kemmer93	145	https://jarret.name	48

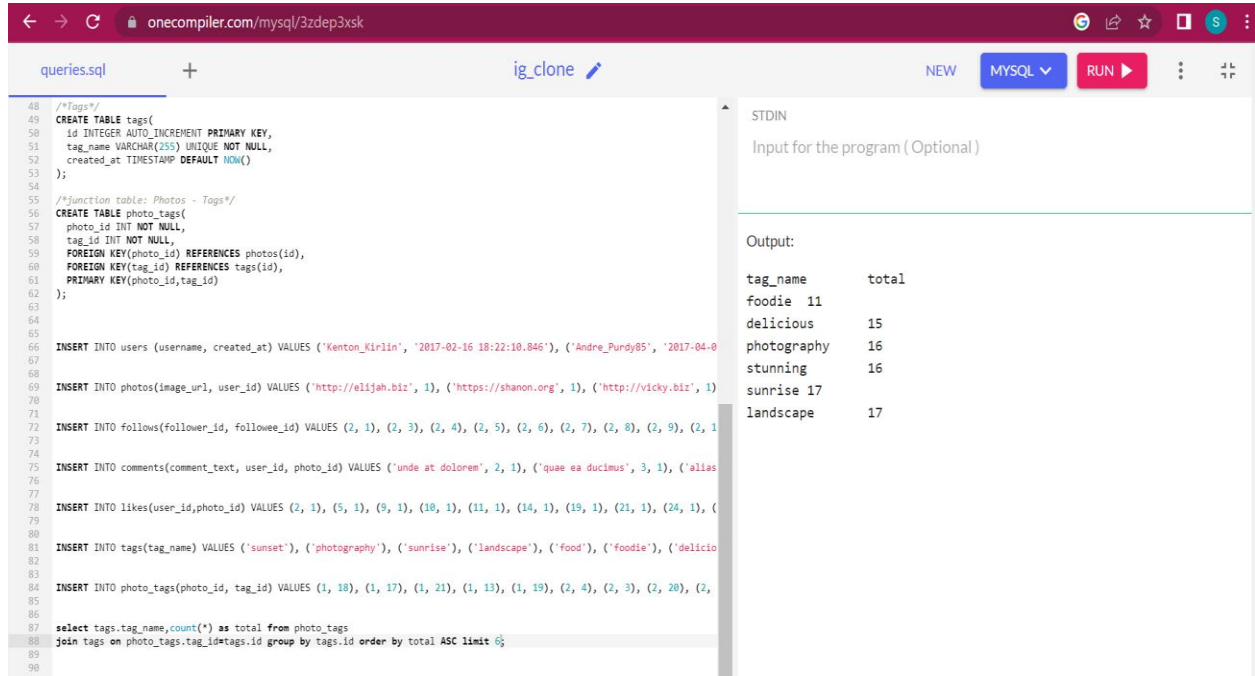
**4. Hash tag Researching:** A partner brand wants to know, which hash tags to use in the post to reach the most people on the platform. Identify and suggest the top 5 most commonly used hash tags on the platform.

SELECT tags.tag\_name,count(\*) as total FROM photo\_tags

join tags on photo\_tags.tag\_id=tags.id

GROUP BY tags.id

ORDER BY total ASC LIMIT 6;



The screenshot shows a web-based MySQL IDE interface. The left pane contains a SQL script with the following queries:

```
48 /*tags*/
49 CREATE TABLE tags(
50   id INTEGER AUTO_INCREMENT PRIMARY KEY,
51   tag_name VARCHAR(255) UNIQUE NOT NULL,
52   created_at TIMESTAMP DEFAULT NOW()
53 );
54
55 /*junction table: Photos - Tags*/
56 CREATE TABLE photo_tags(
57   photo_id INT NOT NULL,
58   tag_id INT NOT NULL,
59   FOREIGN KEY(photo_id) REFERENCES photos(id),
60   FOREIGN KEY(tag_id) REFERENCES tags(id),
61   PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65 INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.845'), ('Andre_Purdy85', '2017-04-0
66
67
68
69 INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1)
70
71
72 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
73
74
75 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolore', 2, 1), ('quae ea ducimus', 3, 1), ('alias
76
77
78 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
79
80
81 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
82
83
84 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
85
86
87 select tags.tag_name,count(*) as total from photo_tags
88 join tags on photo_tags.tag_id=tags.id group by tags.id order by total ASC limit 6;
```

The right pane shows the output of the query:

```
STDIN
Input for the program ( Optional)

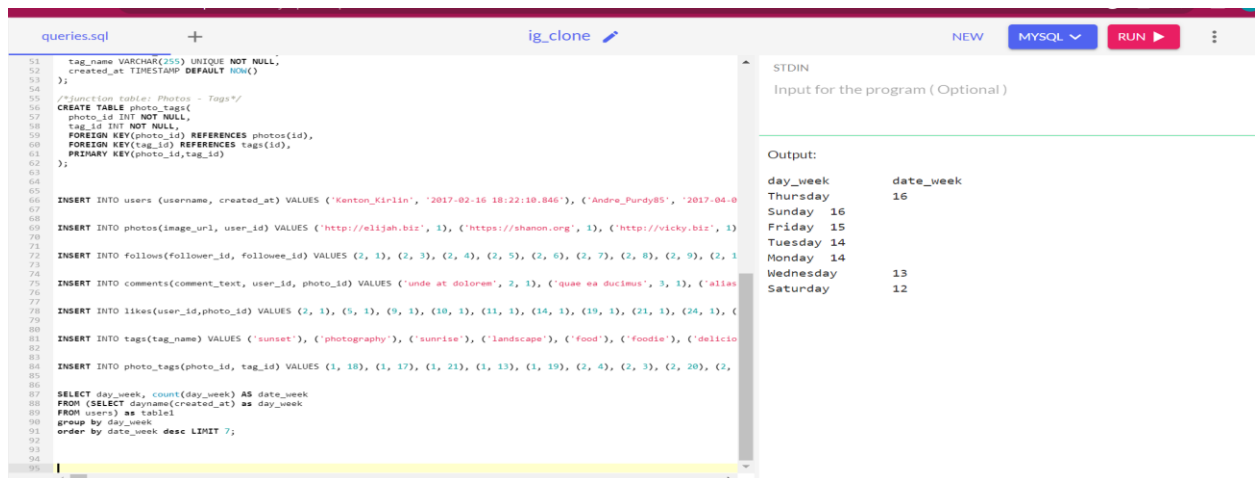
Output:

tag_name      total
foodie        11
delicious     15
photography   16
stunning      16
sunrise       17
landscape     17
```

tag_name	total
foodie	11
delicious	15
photography	16
stunning	16
sunrise	17
landscape	17

5. **Launch AD Campaign:** The team wants to know, which day would be the best day to launch ADs. What day of the week do most users register on? Provide insights on when to schedule an ad campaign.

```
SELECT day_week, count(day_week) AS date_week
FROM (SELECT dayname(created_at) AS day_week
FROM users) as table1
GROUP BY day_week
ORDER BY date_week DESC LIMIT 7;
```



The screenshot shows a MySQL query editor with a SQL script in the left pane and its output in the right pane. The script includes database setup, table creation, and data insertion. The final query is a grouped count of users by day of the week, ordered by count in descending order.

```
queries.sql + ig_clone NEW MYSQL RUN
```

```
51 tag_name VARCHAR(255) UNIQUE NOT NULL,
52 created_at TIMESTAMPT DEFAULT NOW()
53 );
54
55 --function table: Photos - Tags--
56 CREATE TABLE photo_tags(
57 photo_id INT NOT NULL,
58 tag_id INT NOT NULL,
59 FOREIGN KEY(photo_id) REFERENCES photos(id),
60 FOREIGN KEY(tag_id) REFERENCES tags(id),
61 PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65
66 INSERT INTO users (username, created_at) VALUES ('Kanton_Kirilin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-0
67
68
69 INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shamon.org', 1), ('http://vicky.biz', 1)
70
71
72 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
73
74
75 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolore', 2, 1), ('quae ea ducinus', 3, 1), ('alias
76
77
78 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
79
80
81 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
82
83
84 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
85
86
87 SELECT day_week, count(day_week) AS date_week
88 FROM (SELECT dayname(created_at) as day_week
89 FROM users) as table1
90 GROUP BY day_week
91 order by date_week desc LIMIT 7;
92
93
94
95
```

Output:

day_week	date_week
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

day_week	date_week
Thursday	16
Sunday	16
Friday	15
Tuesday	14

Monday	14
Wednesday	13
Saturday	12

Thursdays and Sundays are two days with most user signup on Instagram.

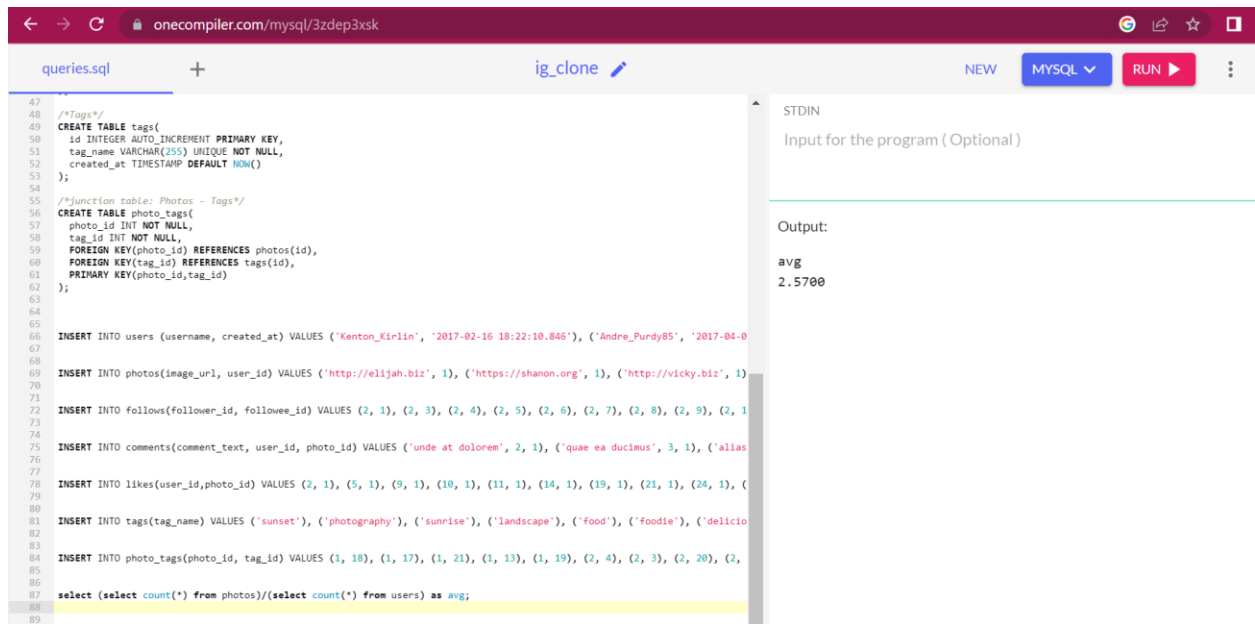
**B) Investor Metrics:** Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds

1. **User Engagement:** Are users still as active and post on Instagram or they are making fewer posts. Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users.

**Total number of photos on Instagram / Total number of users**

```
SELECT (SELECT count(*) FROM photos)/(SELECT count(*) FROM users) AS
average;
```





The screenshot shows the onecompiler.com MySQL interface. The left pane contains a SQL script with the following content:

```
47
48 /*Tags*/
49 CREATE TABLE tags(
50   id INTEGER AUTO_INCREMENT PRIMARY KEY,
51   tag_name VARCHAR(255) UNIQUE NOT NULL,
52   created_at TIMESTAMP DEFAULT NOW()
53 );
54
55 /*function tables: Photos - Tags*/
56 CREATE TABLE photo_tags(
57   photo_id INT NOT NULL,
58   tag_id INT NOT NULL,
59   FOREIGN KEY(photo_id) REFERENCES photos(id),
60   FOREIGN KEY(tag_id) REFERENCES tags(id),
61   PRIMARY KEY(photo_id,tag_id)
62 );
63
64
65
66 INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-0
67
68
69 INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1)
70
71
72 INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
73
74
75 INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolorem', 2, 1), ('que ea ducimus', 3, 1), ('alias
76
77
78 INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
79
80
81 INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
82
83
84 INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
85
86
87 select (select count(*) from photos)/(select count(*) from users) as avg;
88
89
```

The right pane shows the output of the query:

```
STDIN
Input for the program ( Optional)

Output:
avg
2.5700
```

average
2.5700

**2. Bots & Fake Accounts:** The investors want to know if the platform is crowded with fake and dummy accounts. 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, username, count(*) AS count_likes FROM users
inner join likes on users.id = likes.user_id
GROUP BY likes.user_id
HAVING count_likes = (SELECT count(*) FROM photos) ;
```

The screenshot shows a MySQL command-line interface with a dark theme. The top bar displays navigation icons, the URL 'onecompiler.com/mysql/3zdep3xsk', and window controls. The main window is divided into three panes:

- Left Pane (queries.sql):** Contains SQL code for creating tables and inserting data. The code includes:
 

```

      /*Tags*/
      CREATE TABLE tags(
        id INTEGER AUTO_INCREMENT PRIMARY KEY,
        tag_name VARCHAR(255) UNIQUE NOT NULL,
        created_at TIMESTAMP DEFAULT NOW()
      );

      /*Junction table: Photos - Tags*/
      CREATE TABLE photo_tags(
        photo_id INT NOT NULL,
        tag_id INT NOT NULL,
        FOREIGN KEY(photo_id) REFERENCES photos(id),
        FOREIGN KEY(tag_id) REFERENCES tags(id),
        PRIMARY KEY(photo_id,tag_id)
      );

      INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-0
      INSERT INTO photos (image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1)
      INSERT INTO follows (follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 1
      INSERT INTO comments (comment_text, user_id, photo_id) VALUES ('unde at dolorem', 2, 1), ('quae ea ducimus', 3, 1), ('alias
      INSERT INTO likes (user_id, photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (
      INSERT INTO tags (tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('delicio
      INSERT INTO photo_tags (photo_id, tag_id) VALUES (1, 16), (1, 17), (1, 21), (1, 13), (1, 19), (2, 4), (2, 3), (2, 20), (2,
      select user_id, username, count(*) as count_likes from users
      inner join likes on users.id = likes.user_id
      group by likes.user_id having count_likes = (select count(*) from photos);
      
```
- Middle Pane (ig\_clone):** Contains the text 'ig\_clone' and a pencil icon.
- Right Pane:** Shows the output of the SQL queries. It includes a 'STDIN' section with the text 'Input for the program (Optional)' and an 'Output:' section displaying a table of user likes:
 

user_id	username	count_likes
5	Aniya_Hackett	257
14	Jaclyn81	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

## RESULT

By this Instagram user analysis project I had gained a knowledge about the dataset being analyzed, and the queries and techniques employed. The interpretation and action ability of the results would also depend.

## DRIVE LINK



[https://drive.google.com/file/d/10-vCx1VXrsek946TH1n2bWMbeyWHhGEV/view?usp=drive\\_link](https://drive.google.com/file/d/10-vCx1VXrsek946TH1n2bWMbeyWHhGEV/view?usp=drive_link)