

NAME: CHETANA

## Instagram User Analytics

### A) Marketing Analysis:

1. **Loyal User Reward:** The marketing team wants to reward the most loyal users, i.e., those who have been using the platform for the longest time.

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

Query 1 x

Limit to 1000 rows

```
114 ('Bethany20', '2016-06-03 23:31:53.322'), ('Frederik_Rice', '2016-07-06 21:56:28.654'),
115 ('Willie_Leuschke', '2017-02-15 01:40:53.310'), ('Damon35', '2016-10-31 14:44:27.239'),
116 ('Nicole71', '2016-05-09 17:30:22.371'), ('Keenan.Schamberger60', '2016-08-28 14:57:28.221'),
117 ('Tomas.Beatty93', '2017-02-11 11:38:55.026'), ('Imani_Nicolas17', '2017-01-31 22:59:34.108'),
118 ('AleK_Watsica', '2016-12-10 07:43:58.083'), ('Javonte83', '2017-03-27 22:06:37.433');
119
120 • select * from users
121 order by created_at asc;
```

Result Grid

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
71	Nia_Haag	2016-05-14 15:38:50
40	Rafael.Hilde2	2016-05-19 09:51:26
58	Aurelie71	2016-05-31 06:20:57
88	Clint27	2016-06-02 21:40:10
91	Bethany20	2016-06-03 23:31:53
26	Josianne.Friesen	2016-06-07 12:47:01
39	Kelsi26	2016-06-08 17:48:08
73	Jaylan.Lakin	2016-06-10 23:58:52
9	Gus93	2016-06-24 19:36:31
69	Karley_Bosco	2016-06-24 23:38:52
48	Granville_Kutth	2016-06-26 03:10:22
66	Mike.Auer39	2016-07-01 17:36:15
92	Frederik_Rice	2016-07-06 21:56:29
44	Seth46	2016-07-07 11:40:27
34	Pearl7	2016-07-08 21:42:01

users 1 x

Output

Action Output

#	Time	Action	Message
2	12:45:49	USE ig_clone	0 row(s) affected
3	12:45:54	CREATE TABLE users(id INT AUTO_INCREMENT UNIQUE PRIMARY KEY, username VARCHAR(255) NOT NULL, created_at TIMESTAMP DEFA...	0 row(s) affected
4	12:48:27	CREATE TABLE photos(id INT AUTO_INCREMENT PRIMARY KEY, image_url VARCHAR(355) NOT NULL, user_id INT NOT NULL, created_at TI...	0 row(s) affected
5	12:48:33	CREATE TABLE comments(id INT AUTO_INCREMENT PRIMARY KEY, comment_text VARCHAR(255) NOT NULL, user_id INT NOT NULL, photo_j...	0 row(s) affected

3 • select \* from users

4 order by created\_at asc

5 LIMIT 5;

Result Grid

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

*Five oldest users are:*

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 7:56:26

Command:

**select \* from users  
order by created\_at asc;  
LIMIT 5;**

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

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

Query 1

```

247 ('https://helmer.org', 96), ('https://maggie.info', 96),
248 ('https://cecilia.net', 96), ('http://ayla.org', 97),
249 ('https://elyssa.biz', 97), ('http://jennie.com', 98),
250 ('http://ryleigh.info', 99), ('https://darien.name', 99),
251 ('https://xzavier.org', 99), ('https://kaela.name', 100),
252 ('http://dedrick.info', 100);
253
254 • SELECT username
255 FROM users
256 LEFT JOIN photos ON users.id = photos.user_id
257 WHERE photos.user_id IS NULL;

```

Result Grid

username
Aniya_Hackett
Kassandra_Homenick
Jacyn81
Rodo33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

Result 2

Output

Action Output

#	Time	Action	Message
7	12:49:08	CREATE TABLE follows(follower_id INT NOT NULL, followee_id INT NOT NULL, created_at TIMESTAMP DEFAULT NOW(), FOREIGN KEY (followee_id) REFERENCES users(id))	0 row(s) affected

Users who never posted their photos are:

Aniya\_Hackett

Kasandra_Homenick
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

*Command:*

```
SELECT username
FROM users
LEFT JOIN photos ON users.id = photos.user_id
WHERE photos.user_id IS NULL;
```

- 
3. **Contest Winner Declaration:** The team has organized a contest where the user with the most likes on a single photo wins.

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

Query 1

```

320 (40, 16), (41, 16), (42, 16), (43, 16), (44, 16), (47, 16), (52, 16), (54, 16),
321 (56, 16), (57, 16), (60, 16), (61, 16), (66, 16), (69, 16), (71, 16), (72, 16),
322 (73, 16), (75, 16), (76, 16), (91, 16), (93, 16), (95, 16), (96, 16), (100, 16),
323 (4, 17), (5, 17), (11, 17), (14, 17), (16, 17), (21, 17), (24, 17), (27, 17), (28, 17),
324 (30, 17), (36, 17), (37, 17), (41, 17), (44, 17), (48, 17), (52, 17), (54, 17), (55, 17),
325 (57, 17), (60, 17), (61, 17), (63, 17), (65, 17), (66, 17), (71, 17), (75, 17), (76, 17),
326 (78, 17), (79, 17), (82, 17), (91, 17), (93, 17), (94, 17), (97, 17), (98, 17), (99, 17),
327 (5, 18), (10, 18), (12, 18), (14, 18), (18, 18), (21, 18), (22, 18), (24, 18), (31, 18),
328 (36, 18), (41, 18), (46, 18), (47, 18), (48, 18), (54, 18), (57, 18), (61, 18), (62, 18),
329 (91, 18), (95, 18), (96, 18), (98, 18), (4, 19), (5, 19), (6, 19), (10, 19), (13, 19),
330 (14, 19), (20, 19), (21, 19), (24, 19), (28, 19), (31, 19), (36, 19), (38, 19), (40, 19);
331
332 • SELECT users.id, users.username, users.created_at, photos.image_url, COUNT(likes.user_id) AS likes_count
333 FROM users
334 JOIN photos ON users.id = photos.user_id
335 LEFT JOIN likes ON photos.id = likes.photo_id
336 GROUP BY users.id, users.username, users.created_at, photos.image_url
337 ORDER BY likes_count DESC
338 LIMIT 1;
339

```

Result Grid

id	username	created_at	image_url	likes_count
2	Andre_Purdy85	2017-04-02 17:11:21	https://selina.name	38

Winner of the contest is:

2	Andre_Purdy85	2017-04-02 17:11:21	https://selina.name	38
---	---------------	---------------------	---------------------	----

Command:

```

SELECT users.id, users.username, users.created_at, photos.image_url, COUNT(likes.user_id)
AS likes_count
FROM users
JOIN photos ON users.id = photos.user_id
LEFT JOIN likes ON photos.id = likes.photo_id
GROUP BY users.id, users.username, users.created_at, photos.image_url
ORDER BY likes_count DESC
LIMIT 1;

```

4. **Hashtag Research:** A partner brand wants to know the most popular hashtags to use in their posts to reach the most people. Your Task: Identify and suggest the top five most commonly used hashtags on the platform.

Query 1

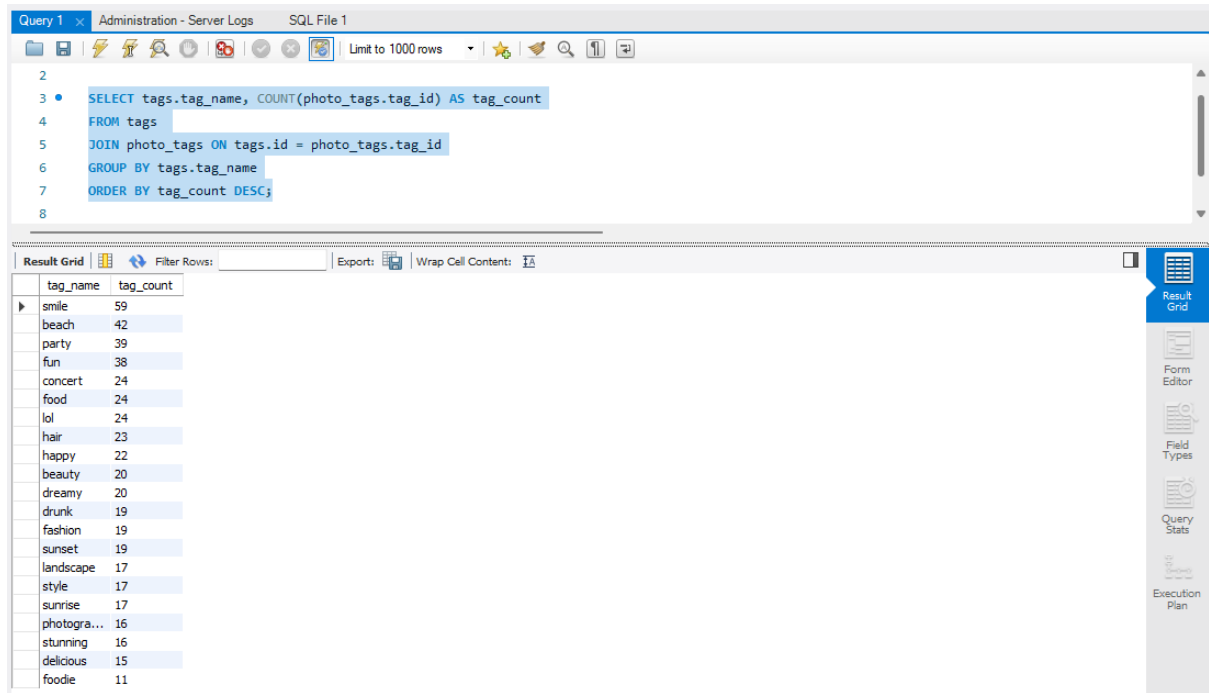
```

3 • SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count
4 FROM tags
5 JOIN photo_tags ON tags.id = photo_tags.tag_id
6 GROUP BY tags.tag_name
7 ORDER BY tag_count DESC
8 LIMIT 5;
9

```

Result Grid

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



Query 1 x Administration - Server Logs SQL File 1

Limit to 1000 rows

```

2
3 • SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count
4 FROM tags
5 JOIN photo_tags ON tags.id = photo_tags.tag_id
6 GROUP BY tags.tag_name
7 ORDER BY tag_count DESC;
8

```

tag_name	tag_count
smile	59
beach	42
party	39
fun	38
concert	24
food	24
lol	24
hair	23
happy	22
beauty	20
dreamy	20
drunk	19
fashion	19
sunset	19
landscape	17
style	17
sunrise	17
photogra...	16
stunning	16
delicious	15
foodie	11

Top 5 commonly used hashtags on the platform:

smile	59
beach	42
party	39
fun	38
concert	24

Command:

```

SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count
FROM tags
INNER JOIN photo_tags ON tags.id = photo_tags.tag_id
GROUP BY tags.tag_name
ORDER BY tag_count DESC
LIMIT 5;

```

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

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

Query 1 Administration - Server Logs SQL File 1

Limit to 1000 rows

```

8      LIMIT 5;
9
10     SELECT DAYNAME(created_at) AS dayy, COUNT(*) AS registration_count
11     FROM users
12     GROUP BY dayy
13     ORDER BY registration_count DESC;
14

```

Result Grid

dayy	registration_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Result 9

Output

#	Time	Action	Message
6	14:39:32	SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count FROM tags JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tags.tag...	5 row(s) returned
7	14:40:00	SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count FROM tags JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tags.tag...	21 row(s) returned
8	14:42:11	SELECT tags.tag_name, COUNT(photo_tags.tag_id) AS tag_count FROM tags JOIN photo_tags ON tags.id = photo_tags.tag_id GROUP BY tags.tag...	5 row(s) returned
9	17:20:08	select * from users LIMIT 0, 1000	100 row(s) returned
10	17:20:41	SELECT DAYNAME(created_at) AS registration_day, COUNT(*) AS registration_count FROM users GROUP BY registration_day ORDER BY registrati...	7 row(s) returned

The day of week when most users register:

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

*Insight: From the above output we can say that peak registrations occur on THURSDAY & SUNDAY, thus it would be optimal to schedule an ad campaign on these days to attract engagement & increased user activity.*

Command:

```

SELECT DAYNAME(created_at) AS dayy, COUNT(*) AS registration_count
FROM users
GROUP BY dayy
ORDER BY registration_count DESC;

```

## 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.

**Your 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.**

The screenshot shows a database query editor with a SQL query in the top pane and its results in the bottom pane. The query is:

```
SELECT COUNT(*) / (SELECT COUNT(*) FROM users) AS avg_post_per_user
FROM photos;
```

The results pane shows a single row with the value 2.5700 for the column avg\_post\_per\_user.

Below the results pane, there is a log of actions with the following entries:

#	Time	Action	Message
11	17:23:46	SELECT DAYNAME(created_at) AS day, COUNT(*) AS registration_count FROM users GROUP BY day ORDER BY registration_count DESC LIMIT ...	7 row(s) returned
12	17:24:15	SELECT DAYNAME(created_at) AS day, COUNT(*) AS registration_count FROM users GROUP BY day ORDER BY registration_count ASC LIMIT 0...	7 row(s) returned
13	17:24:26	SELECT DAYNAME(created_at) AS day, COUNT(*) AS registration_count FROM users GROUP BY day ORDER BY registration_count DESC LIMIT ...	7 row(s) returned

Average No. Of posts per user on Instagram:  
**2.5700**

Command:

**SELECT COUNT(\*) / (SELECT COUNT(\*) FROM users) AS average  
FROM photos;**

**OR**

**--total posts/total users**

**SELECT (SELECT COUNT(\*) FROM photos) / (SELECT COUNT(\*) FROM users) AS  
average;**

2. **Bots & Fake Accounts:** Investors want to know if the platform is crowded with fake and dummy accounts.  
Your Task: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

Query 1 x project 2

```

53 • SELECT user_id, COUNT(*) as num_likes
54 FROM likes
55 GROUP BY user_id
56 HAVING num_likes = (SELECT COUNT(*) FROM photos);
57 • SELECT u.username, COUNT(*) as num_likes
58 FROM users u
59 JOIN likes l ON u.id = l.user_id
60 GROUP BY u.id
61 HAVING num_likes = (SELECT COUNT(*) FROM photos);
62

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

username	num_likes
Aniya_Hackett	257
Jaclyn81	257
Rocio33	257
Maxwell.Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257

Result 74 Result 75 x

Potential bots who've liked every single photo on the site:

Aniya_Hackett	257
Jaclyn81	257
Rocio33	257
Maxwell.Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bethany20	257

Command:

```

select user_id, COUNT(*) AS num_likes
from likes
GROUP BY user_id
HAVING num_likes = (SELECT COUNT(*) FROM photos);
select u.username, COUNT(*) AS num_likes
from users u
JOIN likes l ON u.id = l.user_id
GROUP BY u.id
HAVING num_likes = (SELECT COUNT(*) FROM photos);

```



## **REPORT:**

### **PROJECT DESCRIPTION:**

*In this project I set out to create database ig\_clone from the dataset that would capture everything about user interactions on Instagram including relations like users, likes, follows, photos and tags among others; my goal after which had been achieved was to design & implement a relational database for storage then running some queries via SQL so that we may be able to draw useful conclusions from our findings & complete the given task.*

### **APPROACH:**

*MySQL Workbench was used for the project together with the given dataset. The first step was to create and manage the database by running predefined queries, establishing relationships and populating tables with data. Also, some custom queries were applied where DDL, DML commands, joins, subqueries as well as other SQL techniques were used so as to extract required information and insights.*

### **TECH-STACK USED:**

*Worked on MySQL Workbench (v8.0.30.0) which provides a friendly environment for creating and handling databases, writing and executing SQL queries, visualizing data connections hence thanks to its simplicity in installing, ease of access & gui*

### **INSIGHTS:**

- *Data Relationships: Knowing how different pieces of information relate with each other through different ways of relationship brought about a better understanding on the structure of the dataset.*
- *Data Manipulation and Retrieval Optimization: I made sure I implemented and optimized queries as they are efficient means through which we can get our hands on data as well as change it, this helped in improving the performance generally speaking.*

### **RESULTS:**

*Refined Analytical Skills: SQL writing skills became advanced and were enhanced when more in depth analyses were done through optimizing queries.*

*Practical Work: Understanding database management & data analysis was greatly improved by carrying out various tasks using MySQL Workbench therefore I can comfortably work on other projects professionally going forward.*