

# Instagram User Analytics

(SQL Fundamentals)

## ➤ Project Description:

This project is about the Instagram user analytics. Analytics means the systematic computational analysis of data or statistics. In this project we do analysis of data of Instagram users. There is various data about users like post, likes, followers etc. Using Instagram user analytics can help to improve the strategies, so engagement on platform will increase.

I will do analysis using various SQL queries. With the help of SQL queries, I am going to find various information like top 5 earliest users, post with maximum likes, etc.

## ➤ Approach:

First, I will do schema queries. Schema queries are the queries which create database also insert data in database. After creating database, I will do one by one analysis of given question.

I will focus on to evaluating data and getting more accurate information from the raw data.

## ➤ Tech-Stack Used:

MySQL 8.0

## Insights:

During this project I worked on different commands of SQL. Also, project helps me to gain more knowledge about SQL doing the hands-on. I used joins, sorting functions, aggregate function etc. to get results.

## Result:

### A) Marketing:

1. **Rewarding Most Loyal Users:** People who have been using the platform for the longest time.

### SQL Command:

```
WITH base as
(
  SELECT
    username,
    created_at
  FROM
    ig_clone.users
  ORDER BY
    created_at
  Limit 5
)
SELECT * FROM base
```

### Output:

The screenshot shows the DB Fiddle interface with a MySQL v8.0 database. The Schema SQL section contains the following code:

```
1 CREATE DATABASE ig_clone;
2
3 USE ig_clone;
4
5 /*Users*/
6 CREATE TABLE users(
7   id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
8   username VARCHAR(255) NOT NULL,
9   created_at TIMESTAMP DEFAULT NOW()
10 );
11
12 /*Photos*/
13 CREATE TABLE photos(
14   id INT AUTO_INCREMENT PRIMARY KEY,
15   image url VARCHAR(155) NOT NULL
16 );
```

The Query SQL section contains the following query:

```
2 Your Task: Find the 5 oldest users of the Instagram from the database provided**/
3
4
5 WITH base as
6 (SELECT
7   username,
8   created_at
9 FROM
10  ig_clone.users
11 ORDER BY
12  created_at
13 Limit 5
14 )
15
16 SELECT * FROM base
```

The Results section displays the following data:

username	created_at
Darby_Herzog	2016-05-06 00:14:21
Emilio_Bernier52	2016-05-06 13:04:30
Elenor88	2016-05-08 01:30:41
Nicole71	2016-05-09 17:30:22
Jordyn_Jacobson2	2016-05-14 07:56:26

2. **Remind Inactive Users to Start Posting:** By sending them promotional emails to post their 1st photo.

## SQL Command:

```
SELECT
    username
FROM
    ig_clone.users u
LEFT JOIN
    ig_clone.photos p
ON u.id = p.user_id
WHERE
    p.user_id is NULL
ORDER BY
    u.username;
```

## Output:

The screenshot shows the DB Fiddle interface with a MySQL v8.0 database. The SQL query is as follows:

```
-- **/
12 /*Photos*/
13 CREATE TABLE photos(
14     id INT AUTO_INCREMENT PRIMARY KEY,
15     image_url VARCHAR(355) NOT NULL,
16     user_id INT NOT NULL,
17     created_at TIMESTAMP DEFAULT NOW(),
18     FOREIGN KEY(user_id) REFERENCES users(id)
19 );
20
21 /*Comments*/
22 CREATE TABLE comments(
23     id INT AUTO_INCREMENT PRIMARY KEY,
24     comment_text VARCHAR(255) NOT NULL,
25     user_id INT NOT NULL
26 );
```

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

username
Aniya_Hackett
Bartholome Bernhard
Bethany20
Darby_Herzog
David_Osinski47

- 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.

## Output:

DB Fiddle - SQL Database Playground

Database: MySQL v8.0

Schema SQL

```

1  created_at TIMESTAMP DEFAULT NOW()
2  );
3  /*Photos*/
4  CREATE TABLE photos(
5  id INT AUTO_INCREMENT PRIMARY KEY,
6  image_url VARCHAR(355) NOT NULL,
7  user_id INT NOT NULL,
8  created_at TIMESTAMP DEFAULT NOW(),
9  FOREIGN KEY(user_id) REFERENCES users(id)
10 );
11 /*Comments*/
12 CREATE TABLE comments(
13 id INT AUTO_INCREMENT PRIMARY KEY,
14 content TEXT NOT NULL,
15 user_id INT NOT NULL,
16 created_at TIMESTAMP DEFAULT NOW(),
17 FOREIGN KEY(user_id) REFERENCES users(id)
18 );

```

Query SQL

```

1  /**
2  Question 3: Declaring Contest Winner: The team started a contest and the user who
3  gets the most likes on a single photo will win the contest now they wish to declare
4  the winner.
5  Your Task: Identify the winner of the contest and provide their details to the team
6  */
7  select
8  username
9  from
10 (
11 select
12 likes.photo_id,
13 users.username,
14 count(likes.likan id) as like_nican

```

Results

Query #1 Execution time: 36ms

username
Zack_Kemmer93

DB Fiddle - Crafted with \* by Status200 in the United Kingdom

#### 4. Hashtag Researching: A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.

#### Output:

DB Fiddle - SQL Database Playground

Database: MySQL v8.0

Schema SQL

```

1  created_at TIMESTAMP DEFAULT NOW()
2  );
3  /*Photos*/
4  CREATE TABLE photos(
5  id INT AUTO_INCREMENT PRIMARY KEY,
6  image_url VARCHAR(355) NOT NULL,
7  user_id INT NOT NULL,
8  created_at TIMESTAMP DEFAULT NOW(),
9  FOREIGN KEY(user_id) REFERENCES users(id)
10 );
11 /*Comments*/
12 CREATE TABLE comments(
13 id INT AUTO_INCREMENT PRIMARY KEY,
14 content TEXT NOT NULL,
15 user_id INT NOT NULL,
16 created_at TIMESTAMP DEFAULT NOW(),
17 FOREIGN KEY(user_id) REFERENCES users(id)
18 );

```

Query SQL

```

1  use in the post to reach the most people on the platform.
2  Your Task: Identify and suggest the top 5 most commonly used hashtags on the
3  platform
4  */
5  SELECT
6  t.tag_name,
7  COUNT(p.photo_id) AS num_tags
8  FROM
9  ig_clone.photo_tags p
10 INNER JOIN
11 ig_clone.tags t
12 ON p.tag_id = t.id
13 GROUP BY
14 tag_name

```

Results

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

DB Fiddle - Crafted with \* by Status200 in the United Kingdom

5. **Launch AD Campaign:** The team wants to know, which day would be the best day to launch ADs.

**Output:**

The screenshot shows the DB Fiddle web application interface. On the left, there are input fields for 'Fiddle Title' and 'Fiddle Description'. The 'Schema SQL' section contains the following code:

```
9 created_at TIMESTAMP DEFAULT NOW()
10 );
11
12 /*Photos*/
13 CREATE TABLE photos(
14   id INT AUTO_INCREMENT PRIMARY KEY,
15   image_url VARCHAR(355) NOT NULL,
16   user_id INT NOT NULL,
17   created_at TIMESTAMP DEFAULT NOW(),
18   FOREIGN KEY(user_id) REFERENCES users(id)
19 );
20
21 /*Comments*/
22 CREATE TABLE comments(
23   id INT AUTO_INCREMENT PRIMARY KEY,
```

The 'Query SQL' section contains the following query:

```
1 /**
2  Question 4: Launch AD Campaign: The team wants to know, which day would be the best
3  day to launch ADs.
4  Your Task: What day of the week do most users register on? Provide insights on when to
5  schedule an ad campaign
6  */
7  SELECT
8    WEEKDAY(created_at) AS weekday,
9    COUNT(username) AS num_users
10  FROM
11    ig_clone.users
12  GROUP BY 1 ORDER BY 2 DESC
```

The 'Results' section displays a table with the following data:

weekday	num_users
3	16
6	16
4	15
1	14
0	14

The bottom of the screenshot shows the Windows taskbar with the search bar, task icons, and system clock showing 5:36 PM on 16/10/2022.

## B) Investor Metrics:

1. **User Engagement:** Are users still as active and post on Instagram or they are making fewer posts.

**Output:**

DB Fiddle - SQL Database Playground

Schema SQL

```

1  CREATE TABLE users(
2  id INT AUTO_INCREMENT PRIMARY KEY,
3  created_at TIMESTAMP DEFAULT NOW()
4  );
5
6  /*Photos*/
7  CREATE TABLE photos(
8  id INT AUTO_INCREMENT PRIMARY KEY,
9  image_url VARCHAR(355) NOT NULL,
10 user_id INT NOT NULL,
11 created_at TIMESTAMP DEFAULT NOW(),
12 FOREIGN KEY(user_id) REFERENCES users(id)
13 );
14
15 /*Comments*/
16 CREATE TABLE comments(
17 id INT AUTO_INCREMENT PRIMARY KEY,
18 user_id INT NOT NULL,
19 photo_id INT NOT NULL,
20 comment_text VARCHAR(255) NOT NULL,
21 created_at TIMESTAMP DEFAULT NOW(),
22 FOREIGN KEY(user_id) REFERENCES users(id),
23 FOREIGN KEY(photo_id) REFERENCES photos(id)
24 );

```

Query SQL

```

1  /**
2  Question B-1:User Engagement: Are users still as active and post on Instagram or
3  they are making fewer posts
4  Your Task: Provide how many times does average user posts on Instagram. Also,
5  provide the total number of photos on Instagram/total number of users
6  */
7
8  WITH CTE AS
9  (
10 SELECT
11 u.id AS userid,
12 COUNT(p.id) AS photoid
13 FROM
14 ig_clone.users u
15 LEFT JOIN
16 ig_clone.photos p
17 ON u.id = p.user_id
18 )
19
20 SELECT
21 CTE.userid,
22 CTE.photoid
23 FROM CTE

```

Results

total_photos	total_users	photos_per_user
257	100	2.5700

DB Fiddle - Crafted with w by Status200 in the United Kingdom

2. **Bots & Fake Accounts:** The investors want to know if the platform is crowded with fake and dummy accounts.
- Result:**

DB Fiddle - SQL Database Playground

Schema SQL

```

1  CREATE TABLE users(
2  id INT AUTO_INCREMENT PRIMARY KEY,
3  created_at TIMESTAMP DEFAULT NOW()
4  );
5
6  /*Photos*/
7  CREATE TABLE photos(
8  id INT AUTO_INCREMENT PRIMARY KEY,
9  image_url VARCHAR(355) NOT NULL,
10 user_id INT NOT NULL,
11 created_at TIMESTAMP DEFAULT NOW(),
12 FOREIGN KEY(user_id) REFERENCES users(id)
13 );
14
15 /*Comments*/
16 CREATE TABLE comments(
17 id INT AUTO_INCREMENT PRIMARY KEY,
18 user_id INT NOT NULL,
19 photo_id INT NOT NULL,
20 comment_text VARCHAR(255) NOT NULL,
21 created_at TIMESTAMP DEFAULT NOW(),
22 FOREIGN KEY(user_id) REFERENCES users(id),
23 FOREIGN KEY(photo_id) REFERENCES photos(id)
24 );

```

Query SQL

```

1  /**
2  Question B-2:Bots & Fake Accounts: The investors want to know if the platform is
3  crowded with fake and dummy accounts
4  Your Task: Provide data on users (bots) who have liked every single photo on the
5  site (since any normal user would not be able to do this).
6  */
7
8  WITH photo_count AS
9  (
10 SELECT
11 user_id,
12 COUNT(photo_id) AS num_like
13 FROM
14 ig_clone.likes
15 GROUP BY
16 user_id
17 )
18
19 SELECT
20 user_id,
21 num_like
22 FROM photo_count

```

Results

user_id	num_like
75	257
21	257
24	257
91	257
36	257

DB Fiddle - Crafted with w by Status200 in the United Kingdom