# Instagram User Analytics

## Trainity Project #2

Instagram

Submission by –

Ananaya Lal

Data Analytics Trainee

# Project Description

- User analytics form a core part of every digital product, be it a mobile application, software, or website. It enables a wider understanding of user behavior based on interaction with the product, which provides valuable insights, and consequently, helps in effective decision-making.
- In this project, a dataset of users on the Instagram platform is utilized. This dataset is sorted to produce specific output to obtain user interactions (posts, likes, comments, etc.) and derive meaningful insights.
- These insights are then analyzed from a marketing and investment point-of-view. In doing so, parameters like user loyalty, user engagement, hashtag usage, and bot identification are found. These insights will help in user targeting, deciding what ad campaigns to run and when, and eliminating bots for a leaner, improved marketing and investment strategy.

# Approach

- This project starts by first obtaining a sample dataset of users and their interactions on the Instagram platform.
- This data is then cleaned and stored in a database called *ig*_clone, using multiple tables such as *comments*, *follows*, *likes*, *photo_tags*, *photos*, *tags*, and *users*.
- Once this is done, SQL queries are executed on these tables to obtain specific data as per requirements.
- In the execution of this project, the following software was used:
- 1. MySQL v8.0.33
- → It was used to run SQL queries for creating the database, performing operations on the tables within the database, and obtaining desired outputs.

# Insights

- *From a Marketing department's view*:
- 1. Identifying the 5 oldest users on the Instagram platform based on the available dataset to reward them for their loyalty
- 2. Identifying users who have not posted a single photo on the Instagram platform to send promotional emails reminding them to start posting
- 3. Identifying the user who gets the most likes on a single photo to be declared the winner of a new contest
- 4. Identifying the 5 most used hashtags on the Instagram platform so as to use them for expanded reach
- 5. Identifying the day(s) of the week when most users register on the Instagram platform in order to schedule an ad campaign accordingly.

# Insights

- *From an Investor end* :
- 1. Identifying average posts per user on the Instagram platform to understand if users are active
- 2. Identifying fake accounts/bots to clean up and streamline target audience

# Identifying the 5 oldest users

The results show that users '**Darby_Herzog**', '**Emilio_Bernier52**', '**Elenor88**', '**Nicole71**', and '**Jordyn.Jacobson2**' are the oldest users in that order on the platform.

```
mysql> USE ig_clone;
Database changed
mysql> select * from users
    -> Order by date(created_at)
    -> limit 5;
+----+------------------+---------------------+
| id | username         | created_at          |
+----+------------------+---------------------+
| 80 | Darby_Herzog     | 2016-05-06 00:14:21 |
| 67 | Emilio_Bernier52 | 2016-05-06 13:04:29 |
| 63 | Elenor88         | 2016-05-08 01:30:40 |
| 95 | Nicole71         | 2016-05-09 17:30:22 |
| 38 | Jordyn.Jacobson2 | 2016-05-14 07:56:25 |
+----+------------------+---------------------+
5 rows in set (0.02 sec)
```

# Identifying users who have not posted a single photo

The results show the user IDs, usernames, and profile creation dates of **26** users who have not posted a single photo on the platform

```
mysql> select * from users
    -> left join photos
    -> on users.id = photos.user_id
    -> where user_id is NULL;
```

| id | username | created_at | id | image_url | user_id | created_dat |
|---|---|---|---|---|---|---|
| 5 | Aniya_Hackett | 2016-12-07 01:04:39 | NULL | NULL | NULL | NULL |
| 7 | Kasandra_Homenick | 2016-12-12 06:50:07 | 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 02:32:43 | NULL | NULL | NULL | NULL |
| 25 | Tierra.Trantow | 2016-10-03 12:49:20 | NULL | NULL | NULL | NULL |
| 34 | Pearl7 | 2016-07-08 21:42:00 | NULL | NULL | NULL | NULL |
| 36 | Ollie_Ledner37 | 2016-08-04 15:42:20 | NULL | NULL | NULL | NULL |
| 41 | Mckenna17 | 2016-07-17 17:25:44 | 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 07:49:33 | NULL | NULL | NULL | NULL |
| 54 | Duane60 | 2016-12-21 04:43:37 | NULL | NULL | NULL | NULL |
| 57 | Julien_Schmidt | 2017-02-02 23:12:48 | NULL | NULL | NULL | NULL |
| 66 | Mike.Auer39 | 2016-07-01 17:36:14 | NULL | NULL | NULL | NULL |
| 68 | Franco_Keebler64 | 2016-11-13 20:09:26 | NULL | NULL | NULL | NULL |
| 71 | Nia_Haag | 2016-05-14 15:38:50 | NULL | NULL | NULL | NULL |
| 74 | Hulda.Macejkovic | 2017-01-25 17:17:27 | NULL | NULL | NULL | NULL |
| 75 | Leslie67 | 2016-09-21 05:14:01 | NULL | NULL | NULL | NULL |
| 76 | Janelle.Nikolaus81 | 2016-07-21 09:26:09 | NULL | NULL | NULL | NULL |
| 80 | Darby_Herzog | 2016-05-06 00:14:21 | NULL | NULL | NULL | NULL |
| 81 | Esther.Zulauf61 | 2017-01-14 17:02:33 | NULL | NULL | NULL | NULL |
| 83 | Bartholome.Bernhard | 2016-11-06 02:31:23 | NULL | NULL | NULL | NULL |
| 89 | Jessyca_West | 2016-09-14 23:47:04 | NULL | NULL | NULL | NULL |
| 90 | Esmeralda.Mraz57 | 2017-03-03 11:52:27 | NULL | NULL | NULL | NULL |
| 91 | Bethany20 | 2016-06-03 23:31:53 | NULL | NULL | NULL | NULL |

# Identifying the user with the most likes on a single photo

The results show that user '**Zack_Kemmer93**' with user ID **52** posted a photo with ID **145** on **13/06/2023** at **9:21 pm** that has received **48** likes, the highest on the platform

```
mysql> use ig_clone;
Database changed
mysql> 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 10;
+------------------+-----+----------------------+-------------+
| username         | id  | image_url            | total_likes |
+------------------+-----+----------------------+-------------+
| Zack_Kemmer93    |  52 | https://hershel.com  |          41 |
| Kaley9           |  30 | http://kenny.com     |          41 |
| Jayson65         |  61 | https://dejon.name   |          41 |
| Alexandro35      |  13 | https://fred.com     |          40 |
| Tomas.Beatty93   |  97 | https://carolanne.com|          40 |
| Javonte83        | 100 | https://brook.com    |          39 |
| Mike.Auer39      |  66 | http://lionel.net    |          39 |
| Seth46           |  44 | http://golden.org    |          39 |
| Ressie_Stanton46 |  62 | https://rigoberto.net|          39 |
| Kasandra_Homenick|   7 | https://selina.name  |          38 |
+------------------+-----+----------------------+-------------+
```

# Identifying the 5 most used hashtags

The results show that '**smile**', '**beach**', '**party**', '**fun**', and '**concert**' are the 5 most used hashtags in that order on the platform

```
mysql> use ig_clone;
Database changed
mysql> select photo_tags.tag_id, tags.tag_name, count(tag_id) as total_use
    -> from photo_tags
    -> join tags
    -> on photo_tags.tag_id = tags.id
    -> group by photo_tags.tag_id
    -> order by total_use desc
    -> limit 5;
+--------+----------+-----------+
| tag_id | tag_name | total_use |
+--------+----------+-----------+
|     21 | smile    |        59 |
|     20 | beach    |        42 |
|     17 | party    |        39 |
|     13 | fun      |        38 |
|     18 | concert  |        24 |
+--------+----------+-----------+
5 rows in set (0.00 sec)
```

# Identifying the busiest days of the week in terms of user registration

The results show that most users register on the platform on **Thursdays** and **Sundays**

```
mysql> use ig_clone;
Database changed
mysql> select dayname(created_at) as week_day, count(*) as frequency
    -> from users
    -> group by week_day
    -> order by frequency desc;
+-----------+-----------+
| week_day  | frequency |
+-----------+-----------+
| Thursday  |        16 |
| Sunday    |        16 |
| Friday    |        15 |
| Tuesday   |        14 |
| Monday    |        14 |
| Wednesday |        13 |
| Saturday  |        12 |
+-----------+-----------+
7 rows in set (0.01 sec)
```

# Identifying average posts per user

The results show that there are a total of **257 posts** on the platform. With **100 total users,** this gives an average of **2.57 posts per user**

Total Posts :

```
mysql> use ig_clone;
Database changed
mysql> select count(id) from photos as total_photos;
+-----------+
| count(id) |
+-----------+
|       257 |
+-----------+
```

Average Post per user

```
mysql> use ig_clone;
Database changed
mysql> select ((select count(*) from photos)/(select count(*) from users)) as avg_post;
+----------+
| avg_post |
+----------+
|   2.5700 |
+----------+
1 row in set (0.01 sec)
```

# Conclusion

- By identifying active, the management team can now focus on targeting the right users. They can utilize campaigns on specific days with specific hashtags to ensure wider reach.
- Moreover, by finding and rewarding loyal users as well as users who secure a high number of likes on their posts, the campaign could be further publicized. It will also ensure continued patronage by existing users while also securing more users, who are attracted by the fact that the platform looks after its regular users. This could, in turn, lead to more posts by more users, thereby increasing the average posts-per-user values.
- This project has been very helpful in understanding basic and advanced SQL concepts, as well as the workings of relational databases. It provides a idea of how product teams work at organizations, how they handle such large amounts of data and come up with the valuable insights.