Project - 2
Instagram User Analytics

Project Description: Perform SQL operations on the given Instagram User Analytics data and find out the answers to the questions asked by the management team.

Approach: Analyse the questions asked and perform the SQL operation according to the need of the management term. We can approach the answers to the questions by running or injecting the correct SQL query.

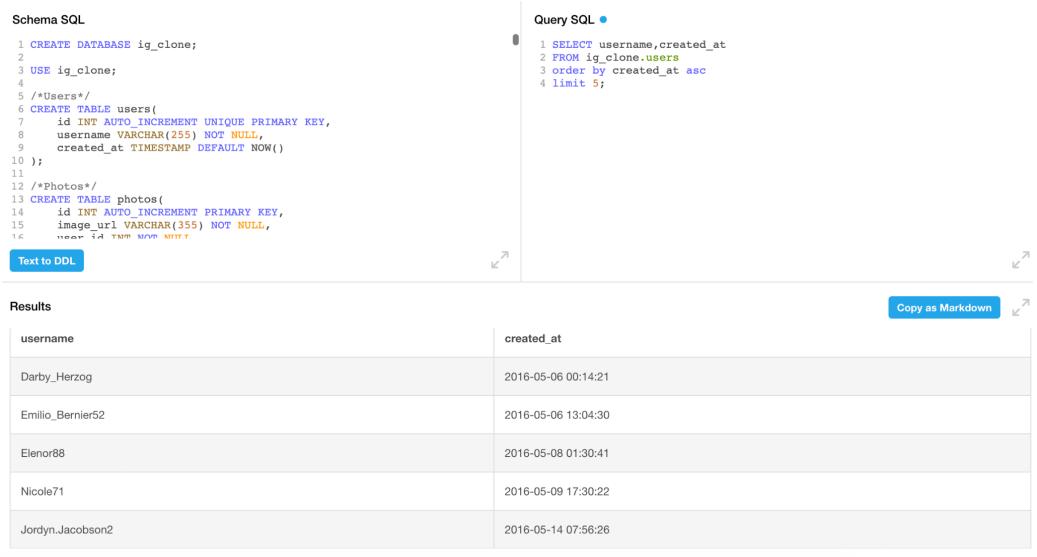
Tech-Stack-Used: https://www.db-fiddle.com/

Insight: By doing this Instagram Analysis i was able to understand the SQL query to be implemented inorder to get the required solution for the problem. The way by which two tables can be compared or connected together in order to perform the operations.

Results:

- **Create a Database**

- **Perform Analysis:**
  1. **Find the 5 oldest users of the Instagram from the database provided.**



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. **Find the users who have never posted a single photo on Instagram**

Query SQL

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

username
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

---

3. **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.**

Schema SQL

```
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
    user_id INT NOT NULL,
    created_dat TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
```

Text to DDL

Query SQL

```
SELECT username,photos.id,photos.image_url,COUNT(*) AS total
FROM ig_clone.photos
INNER JOIN ig_clone.likes
ON likes.photo_id=photos.id
INNER JOIN ig_clone.users
ON photos.user_id=users.id
GROUP BY photos.id
ORDER BY total DESC
LIMIT 1;
```

Results

Copy as Markdown

Query #1 Execution time: 14ms

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

4. **Identify and suggest the top 5 most commonly used hashtags on the platform**

Schema SQL

```
/*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)
);
```

Text to DDL

Query SQL

```
SELECT tag_name,COUNT(*) AS total
FROM ig_clone.photo_tags
JOIN ig_clone.tags
ON photo_tags.tag_id=tags.id
GROUP BY tags.id
ORDER BY total DESC
LIMIT 5;
```

Results

Copy as Markdown

tag_name	total
smile	59
beach	42
party	39
fun	38
food	24

5. **What day of the week do most users register on? Provide insights on when to schedule an ad campaign**

Schema SQL

```sql
CREATE DATABASE ig_clone;

USE ig_clone;

/*Users*/
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
```

Text to DDL

Query SQL

```sql
SELECT DAYNAME(created_at) AS day,COUNT(*) AS total
FROM ig_clone.users
GROUP BY day
ORDER BY total DESC
LIMIT 1;
```

Results

Copy as Markdown

Query #1 Execution time: 0ms

day	total
Thursday	16

- **Investor Metrics**

1. **User Engagement:**
   Provide how many times does average user posts on Instagram.

Schema SQL

```sql
/*Users*/
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
    user_id INT NOT NULL,
    created_dat TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
```

Text to DDL

Query SQL

```sql
SELECT avg (user_id)
FROM ig_clone.photos;
```

Results

Copy as Markdown

Query #1 Execution time: 0ms

avg (user_id)
47.3307

Also, provide the total number of users

Schema SQL

```sql
CREATE DATABASE ig_clone;

USE ig_clone;

/*Users*/
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
```

Text to DDL

Query SQL

```sql
SELECT COUNT(username)
FROM ig_clone.users;
```

Results

Copy as Markdown

Query #1 Execution time: 0ms

COUNT(username)
100

Total number of photos on Instagram

Schema SQL

```sql

/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
    user_id INT NOT NULL,
    created_dat TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);

/*Comments*/
CREATE TABLE comments(
    id INT AUTO_INCREMENT PRIMARY KEY,
    comment_text VARCHAR(255) NOT NULL,
    user_id INT NOT NULL,
    photo id INT NOT NULL,
```

Text to DDL

Query SQL

```sql
SELECT SUM(user_id)
FROM ig_clone.photos;
```

Results

Query #1 Execution time: 0ms

SUM(user_id)
12164

2. Bots & Fake 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).**

Schema SQL

```
    FOREIGN KEY(user_id) REFERENCES users(id),
    FOREIGN KEY(photo_id) REFERENCES photos(id)
);

/*Likes*/
CREATE TABLE likes(
    user_id INT NOT NULL,
    photo_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id),
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    PRIMARY KEY(user_id,photo_id)
);

/*follows*/
```

Text to DDL

Query SQL

```
SELECT users.username, photos.id,photos.image_url,COUNT(*) AS
total_likes
FROM ig_clone.likes
JOIN ig_clone.photos ON photos.id=likes.photo_id
JOIN ig_clone.users ON users.id=likes.photo_id
GROUP BY photos.id
ORDER BY total_likes DESC
LIMIT 5;
```

Results

Copy as Markdown

username	id	image_url	total_likes
Jayson65	61	https://dejon.name	41
Kaley9	30	http://kenny.com	41
Zack_Kemmer93	52	https://hershel.com	41
Tomas.Beatty93	97	https://carolanne.com	40
Alexandro35	13	https://fred.com	40