Darshita Anyawada

SQL Based Instagram Database Clone

Abstract:

This report presents the design and implementation of a relational database model for an Instagram clone application. The database schema captures the fundamental features of the popular social media platform, including user management, photo sharing, commenting, liking, following, and tagging. The report outlines the structure of the database, including the tables and their relationships, along with the key queries used to retrieve and analyze data. Through a series of SQL queries, various insights are derived, such as the registration patterns of users, popular content, user engagement metrics, and hashtag trends. The Instagram clone database serves as a foundational framework for developing social media applications or features, providing a scalable and efficient solution for managing user-generated content and interactions.

Introduction:

In the digital age, social media platforms have become integral to modern communication, connecting individuals across the globe and facilitating the exchange of ideas, experiences, and content. Behind the seamless user experience of these platforms lies robust database management systems (DBMS) and structured query languages (SQL) that power data storage, retrieval, and analysis. This report explores the design and implementation of a relational database model for an Instagram clone application, which seeks to replicate the core functionalities of the renowned social media platform.

Structured Query Language (SQL) serves as the standard language for managing and manipulating relational databases. It provides powerful capabilities for creating, querying, updating, and deleting data within a database management system (DBMS). DBMS, in turn, serves as the software infrastructure that enables efficient storage, retrieval, and management of data.

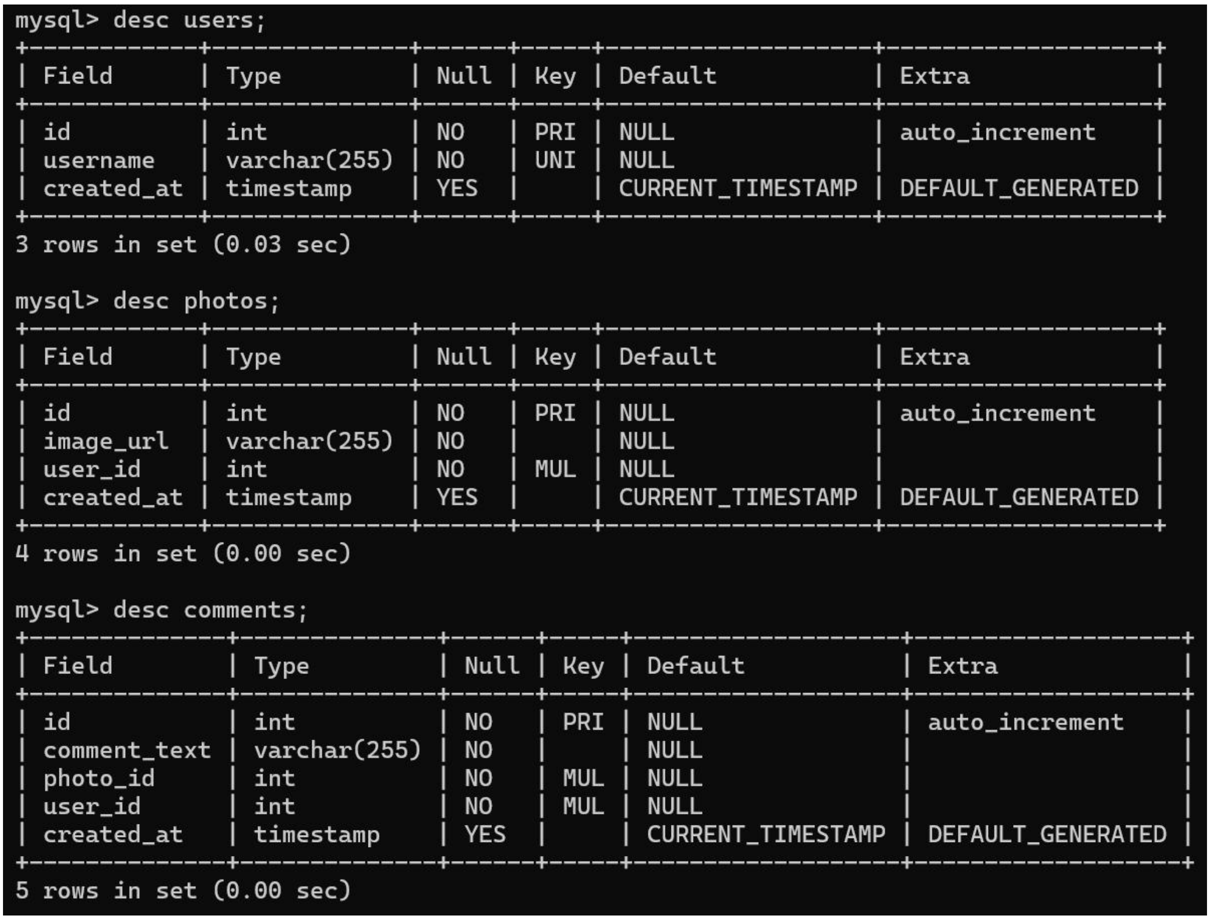
The Instagram clone project involves the creation of a relational database schema that encapsulates various features of the original platform, including user registration, photo sharing, commenting, liking, following, and tagging. Each aspect of the platform's functionality is mapped to corresponding database tables and relationships, ensuring seamless data management and retrieval.

The database schema comprises several interrelated tables, each representing a distinct entity or relationship within the Instagram clone application. These tables include "users," "photos," "comments," "likes," "follows," "tags," and "photo_tags," with each table designed to store specific types of data and facilitate efficient data retrieval through optimized queries.

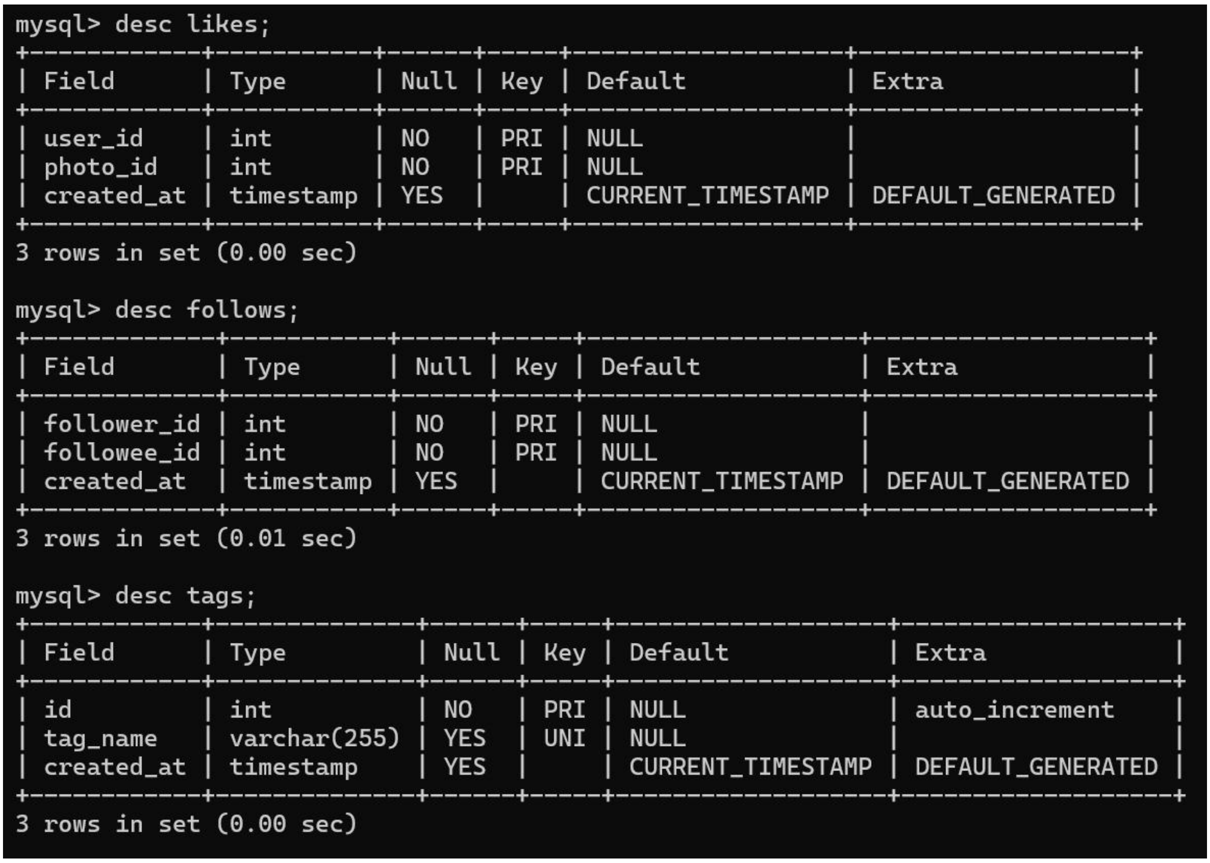
Throughout this report, we showcase various SQL queries utilized to extract meaningful insights from the Instagram clone database. These queries range from identifying the oldest users on the platform to analyzing user engagement metrics, such as post frequency and hashtag trends. Leveraging SQL's capabilities, we gain valuable insights into user behavior, content popularity, and platform dynamics.

Darshita Anyawada

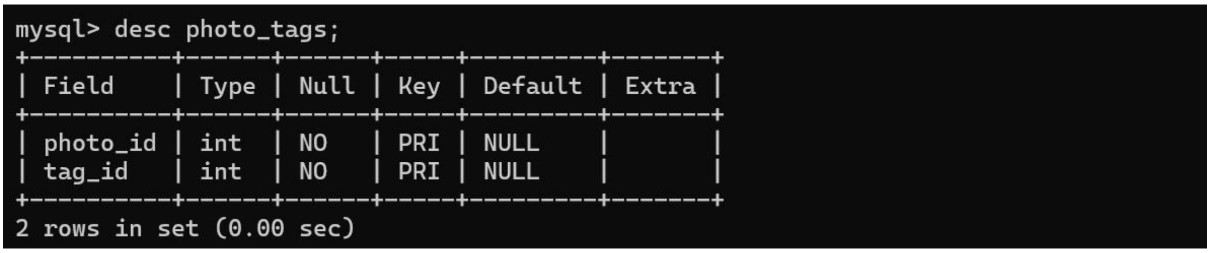
Schema:



```
mysql> desc users;
+------------+--------------+------+-----+-------------------+-------------------+
| Field      | Type         | Null | Key | Default           | Extra             |
+------------+--------------+------+-----+-------------------+-------------------+
| id         | int          | NO   | PRI | NULL              | auto_increment    |
| username   | varchar(255) | NO   | UNI | NULL              |                   |
| created_at | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------+--------------+------+-----+-------------------+-------------------+
3 rows in set (0.03 sec)

mysql> desc photos;
+------------+--------------+------+-----+-------------------+-------------------+
| Field      | Type         | Null | Key | Default           | Extra             |
+------------+--------------+------+-----+-------------------+-------------------+
| id         | int          | NO   | PRI | NULL              | auto_increment    |
| image_url  | varchar(255) | NO   |     | NULL              |                   |
| user_id    | int          | NO   | MUL | NULL              |                   |
| created_at | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------+--------------+------+-----+-------------------+-------------------+
4 rows in set (0.00 sec)

mysql> desc comments;
+--------------+--------------+------+-----+-------------------+-------------------+
| Field        | Type         | Null | Key | Default           | Extra             |
+--------------+--------------+------+-----+-------------------+-------------------+
| id           | int          | NO   | PRI | NULL              | auto_increment    |
| comment_text | varchar(255) | NO   |     | NULL              |                   |
| photo_id     | int          | NO   | MUL | NULL              |                   |
| user_id      | int          | NO   | MUL | NULL              |                   |
| created_at   | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+--------------+--------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```

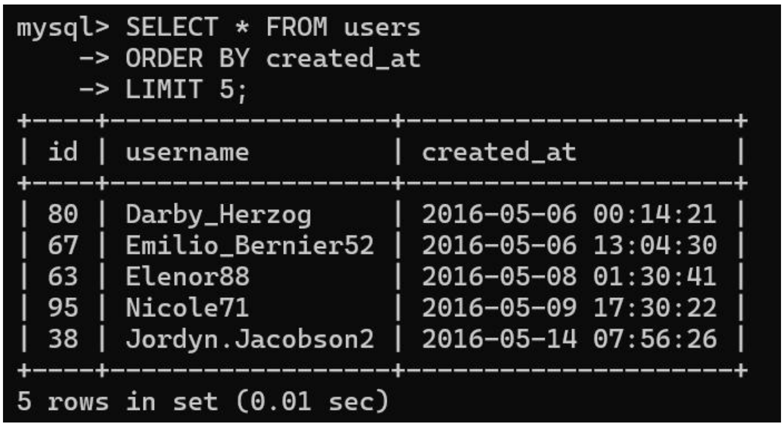


```
mysql> desc likes;
+------------+-----------+------+-----+-------------------+-------------------+
| Field      | Type      | Null | Key | Default           | Extra             |
+------------+-----------+------+-----+-------------------+-------------------+
| user_id    | int       | NO   | PRI | NULL              |                   |
| photo_id   | int       | NO   | PRI | NULL              |                   |
| created_at | timestamp | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------+-----------+------+-----+-------------------+-------------------+
3 rows in set (0.00 sec)

mysql> desc follows;
+-------------+-----------+------+-----+-------------------+-------------------+
| Field       | Type      | Null | Key | Default           | Extra             |
+-------------+-----------+------+-----+-------------------+-------------------+
| follower_id | int       | NO   | PRI | NULL              |                   |
| followee_id | int       | NO   | PRI | NULL              |                   |
| created_at  | timestamp | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-------------+-----------+------+-----+-------------------+-------------------+
3 rows in set (0.01 sec)

mysql> desc tags;
+------------+--------------+------+-----+-------------------+-------------------+
| Field      | Type         | Null | Key | Default           | Extra             |
+------------+--------------+------+-----+-------------------+-------------------+
| id         | int          | NO   | PRI | NULL              | auto_increment    |
| tag_name   | varchar(255) | YES  | UNI | NULL              |                   |
| created_at | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------+--------------+------+-----+-------------------+-------------------+
3 rows in set (0.00 sec)
```

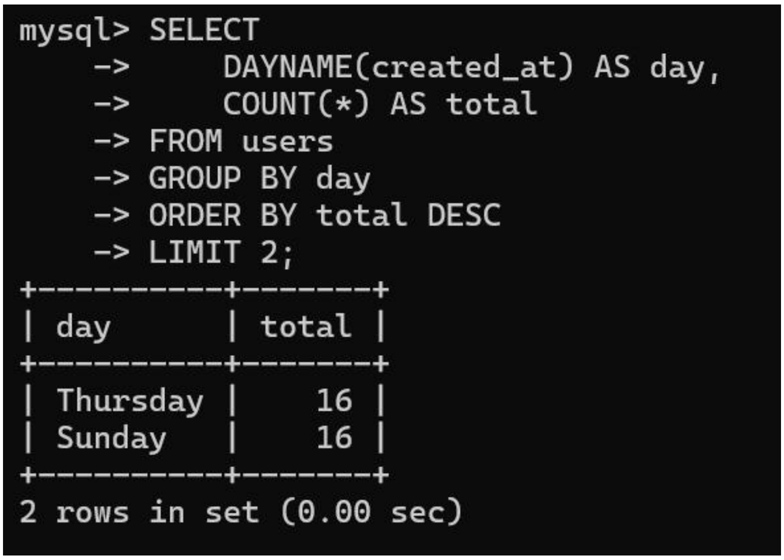


```
mysql> desc photo_tags;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| photo_id | int  | NO   | PRI | NULL    |       |
| tag_id   | int  | NO   | PRI | NULL    |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

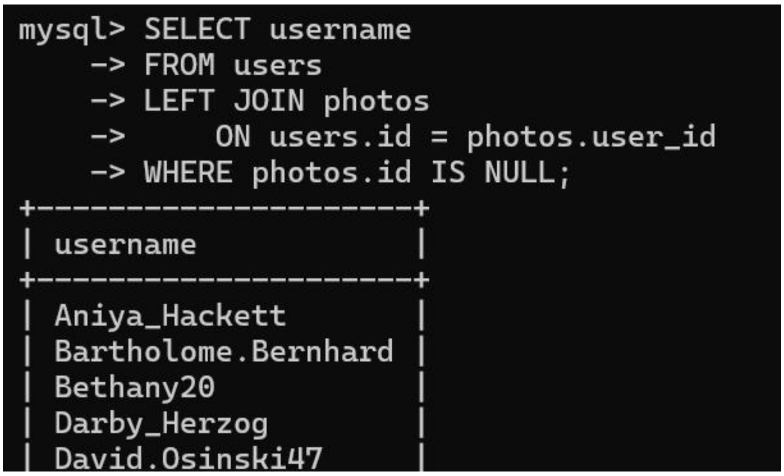
Results:



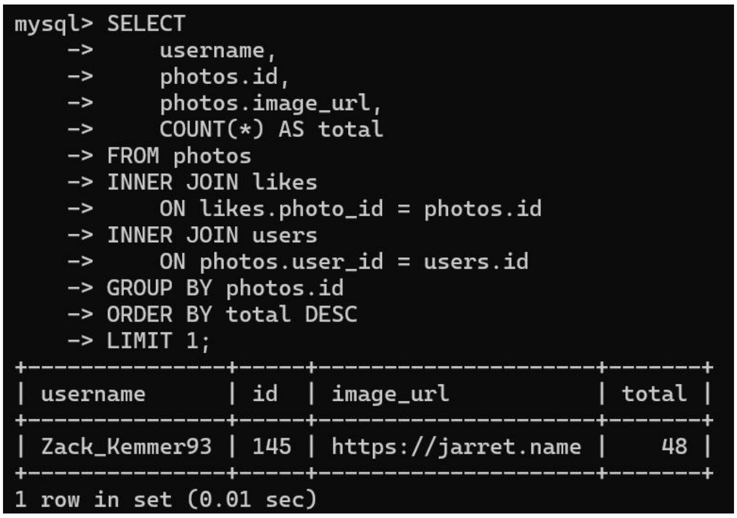
```
mysql> SELECT * FROM users
    -> ORDER BY created_at
    -> LIMIT 5;
+----+------------------+---------------------+
| id | username         | created_at          |
+----+------------------+---------------------+
| 80 | Darby_Herzog     | 2016-05-06 00:14:21 |
| 67 | Emilio_Bernier52 | 2016-05-06 13:04:30 |
| 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 |
+----+------------------+---------------------+
5 rows in set (0.01 sec)
```



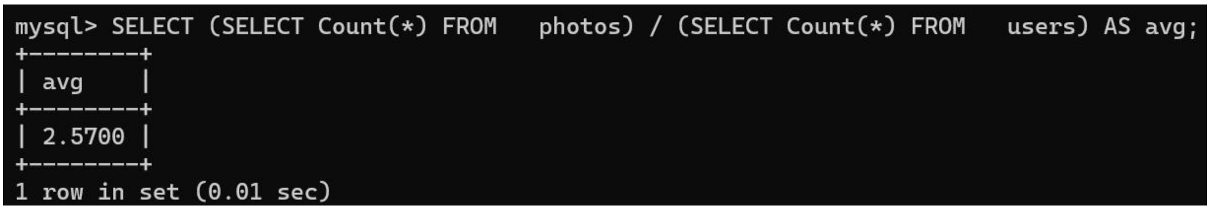
```
mysql> SELECT
    ->     DAYNAME(created_at) AS day,
    ->     COUNT(*) AS total
    -> FROM users
    -> GROUP BY day
    -> ORDER BY total DESC
    -> LIMIT 2;
+----------+-------+
| day      | total |
+----------+-------+
| Thursday |    16 |
| Sunday   |    16 |
+----------+-------+
2 rows in set (0.00 sec)
```



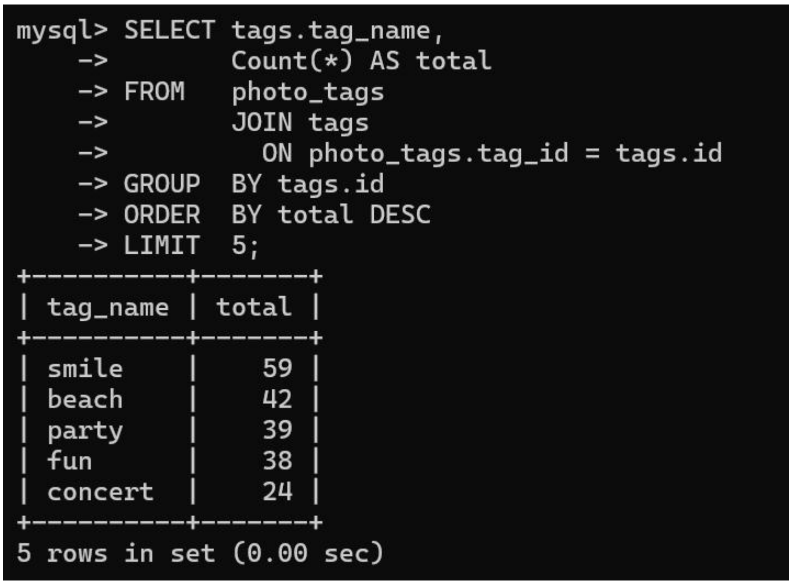
```
mysql> SELECT username
    -> FROM users
    -> LEFT JOIN photos
    ->     ON users.id = photos.user_id
    -> WHERE photos.id IS NULL;
+---------------------+
| username            |
+---------------------+
| Aniya_Hackett       |
| Bartholome.Bernhard |
| Bethany20           |
| Darby_Herzog        |
| David.Osinski47     |
```



```
mysql> 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;
+---------------+-----+---------------------+-------+
| username      | id  | image_url           | total |
+---------------+-----+---------------------+-------+
| Zack_Kemmer93 | 145 | https://jarret.name |    48 |
+---------------+-----+---------------------+-------+
1 row in set (0.01 sec)
```



```
mysql> SELECT (SELECT Count(*) FROM   photos) / (SELECT Count(*) FROM   users) AS avg;
+--------+
| avg    |
+--------+
| 2.5700 |
+--------+
1 row in set (0.01 sec)
```



```
mysql> 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 DESC
    -> LIMIT  5;
+----------+-------+
| tag_name | total |
+----------+-------+
| smile    |    59 |
| beach    |    42 |
| party    |    39 |
| fun      |    38 |
| concert  |    24 |
+----------+-------+
5 rows in set (0.00 sec)
```

Conclusion:

In conclusion, the Instagram clone database project successfully demonstrates the design and implementation of a relational database model to emulate the core functionalities of the original social media platform. Through careful schema design and SQL query execution, we've effectively captured user interactions, content sharing, and engagement metrics within the platform. The insights derived from the database, such as user registration patterns, content popularity, and hashtag trends, offer valuable knowledge for understanding user behaviour and platform dynamics. Overall, this project serves as a solid foundation for developing and enhancing social media applications, showcasing the power of database management systems and structured query languages in facilitating data-driven insights and user experiences.