**Social Media Database Project**

**Overview**

This project models a **social media database** using MySQL, containing information about users, their posts, and comments on the posts. The database demonstrates basic **CRUD operations** (Create, Read, Update, Delete), joins, and aggregations. It consists of three main tables: Users, Posts, and Comments.

**Database Structure**

The project contains **three tables**:

1. **Users**: Stores user information.
   * user\_id: Primary Key, Auto-Increment.
   * username: User's name.
   * email: User's email address.
   * created\_at: Timestamp of account creation.
2. **Posts**: Stores posts created by users.
   * post\_id: Primary Key, Auto-Increment.
   * user\_id: Foreign Key referencing Users.
   * content: Content of the post.
   * created\_at: Timestamp of post creation.
3. **Comments**: Stores comments on posts.
   * comment\_id: Primary Key, Auto-Increment.
   * post\_id: Foreign Key referencing Posts.
   * user\_id: Foreign Key referencing Users.
   * comment: Content of the comment.
   * created\_at: Timestamp of comment creation.

**How to Use**

**1. Creating the Database and Tables**

Run the following SQL script to create the database and tables:

CREATE DATABASE IF NOT EXISTS social\_media;

USE social\_media;

DROP TABLE IF EXISTS Comments, Posts, Users;

CREATE TABLE Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Posts (

post\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

content TEXT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

CREATE TABLE Comments (

comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

post\_id INT NOT NULL,

user\_id INT NOT NULL,

comment TEXT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (post\_id) REFERENCES Posts(post\_id),

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

**2. Inserting Sample Data**

Use the following SQL queries to insert some sample data:

INSERT INTO Users (username, email) VALUES

('Aarav', 'aarav@example.com'),

('Meera', 'meera@example.com'),

('Rohan', 'rohan@example.com'),

('Kavya', 'kavya@example.com'),

('Vivaan', 'vivaan@example.com');

INSERT INTO Posts (user\_id, content) VALUES

(1, 'Excited for the new year!'),

(2, 'Just had a great meal!'),

(3, 'Traveling to Goa next week!'),

(1, 'What are your resolutions?'),

(4, 'Can\'t wait for the festival!'),

(5, 'Just finished a great book!');

INSERT INTO Comments (post\_id, user\_id, comment) VALUES

(1, 2, 'Happy New Year, Aarav!'),

(1, 3, 'Cheers to a great year ahead!'),

(2, 1, 'Sounds delicious, Meera!'),

(3, 4, 'Enjoy your trip to Goa!'),

(4, 5, 'I want to travel too!'),

(5, 1, 'What book did you read?');

**Queries with Purpose**

Here is a list of SQL queries along with their purposes:

1. **View all users:**

SELECT \* FROM Users;

1. **View all posts:**

SELECT \* FROM Posts;

1. **View all comments:**

SELECT \* FROM Comments;

1. **Find posts by a specific user (user\_id = 1):**

SELECT \* FROM Posts WHERE user\_id = 1;

1. **Find comments for a specific post (post\_id = 1):**

SELECT \* FROM Comments WHERE post\_id = 1;

1. **Count the number of posts by each user:**

SELECT user\_id, COUNT(\*) AS post\_count FROM Posts GROUP BY user\_id;

1. **Count the number of comments on each post:**

SELECT post\_id, COUNT(\*) AS comment\_count FROM Comments GROUP BY post\_id;

1. **Find the latest post by each user:**

SELECT user\_id, content, created\_at FROM Posts WHERE (user\_id, created\_at) IN (SELECT user\_id, MAX(created\_at) FROM Posts GROUP BY user\_id);

1. **View posts along with the author's username:**

SELECT p.\*, u.username FROM Posts p JOIN Users u ON p.user\_id = u.user\_id;

1. **Find all comments along with the post content and commenter name:**

SELECT c.\*, u.username, p.content FROM Comments c JOIN Users u ON c.user\_id = u.user\_id JOIN Posts p ON c.post\_id = p.post\_id;

1. **Update user email:**

UPDATE Users SET email = 'aarav123@example.com' WHERE user\_id = 1;

1. **Delete a comment:**

DELETE FROM Comments WHERE comment\_id = 3;

**Limitations**

* **Foreign Key Constraints**: Ensure proper relationships are maintained between tables when deleting or inserting data.
* **Data Loss**: Using DROP or DELETE statements will remove data permanently; handle these with care.
* **Scalability**: This project is intended for learning purposes and may require optimizations for real-world applications.

**Conclusion**

This project demonstrates how to create a simple relational database for a social media platform. It covers key SQL operations, including **joins, aggregations, CRUD operations, and filtering**. You can modify the database further by adding more tables like Likes or Followers to extend its functionality.

**How to Run**

1. Install MySQL Server and MySQL Workbench (or any other client).
2. Open your MySQL client and connect to the server.
3. Copy the SQL code provided in this README and run it.
4. Explore the database using the sample queries.

**Author**

This project was built to explore SQL operations with a **social media database** use case. Feel free to use or modify it as needed for educational purposes!