Mini SQL project : Instagram inspired social media app.

By: Rakshita

June 2025

Introduction :

This project represents a simplified version of a social media application, inspired by platforms like Instagram. It is built entirely using SQL and demonstrates how relational databases work to manage users, posts, comments, and likes. The project includes the creation of multiple related tables, insertion of dummy data, and performing various SQL operations such as JOINs, subqueries, aggregations, and filtering. This hands-on project enhances understanding of real-world database design and manipulation.

Objective :

- To understand and apply core SQL operations using a fictional dataset.
- To demonstrate how data is stored, retrieved, filtered, and related across multiple tables.
- To practice working with SQL queries like SELECT, INSERT, DELETE, UPDATE, ALTER, JOIN, GROUP BY, and subqueries.
- To simulate a working backend for a social media application.

Tools used in this project :

- SQLite / SQL : Writing and executing queries
- Text Editor : Writing the script
- GitHub : Hosting and sharing the project
- LinkedIn : Sharing the project professionally
- Fictional Data : Dummy data for users, posts, likes, and comments

SQL tools used in this project :

- CREATE TABLE
- INSERT INTO
- SELECT
- WHERE
- UPDATE
- ALTER TABLE
- DELETE
- ORDER BY
- GROUP BY
- HAVING
- JOIN
- LEFT OUTER JOIN
- INNER JOIN
- SUBQUERY
- CASE
- AVG()
- MIN()
- MAX()
- COUNT()
- AS

Code :

```
--This data is based on a fictional instagram-like app.
-- All records are self-generated for learning and project purpose by rakshita.
CREATE TABLE user (
id INTEGER PRIMARY KEY AUTOINCREMENT,
user_name TEXT,
gender TEXT,
email TEXT,
join_date TEXT
);

CREATE TABLE posts (
id INTEGER PRIMARY KEY AUTOINCREMENT,
user_id INTEGER,
caption TEXT,
post_date TEXT
);

CREATE TABLE comments (
id INTEGER PRIMARY KEY AUTOINCREMENT,
post_id INTEGER,
user_id INTEGER,
comment TEXT,
comment_date TEXT
);

CREATE TABLE likes (
id INTEGER PRIMARY KEY AUTOINCREMENT,
post_id INTEGER,
user_id INTEGER,
like_date TEXT
);
```

```sql
-- Inserting data in user table.
INSERT INTO user (user_name, gender, email, join_date) VALUES ("Rakshita",
"Female", "Rakshi12@mail.com", "12-08-2021"), ("Shreya", "Female",
"shreya34@mail.com", "09-03-2023"), ("Rohan", "Male",
"rohan45@mail.com", "21-09-2020"), ("Harshita", "Female",
"harshi866@mail.com", "30-05-2018"), ("Priyanshu", "Male",
"priyanshu@123mail.com", "04-07-2022"), ("Vaibhav", "Male",
"vaibhu482@mail.com", "18-01-2022");

-- Inserting data in posts table.
INSERT INTO posts (user_id, caption, post_date) VALUES(2, "Self care is
important..", "03-05-2023"),(1, "Stitched with love and extra care..", "07-11-
2021"),(5, "Stay fit to be hit.. GYM VIBES", "17-11-2022"),(4, "Coffee +
sunrise = peaceful morning..", "13-06-2018"),(3, "In a way to find peace..",
"01-01-2021"),(4, "Laughing until my hearts feels light..", "02-04-2019");

-- Inserting data in comments table.
INSERT INTO comments (post_id, user_id, comment, comment_date)
VALUES(1, 3,"Self-love, queen...", "06-05-2023"),(2, 4, "Love your new kurti
design..", "08-11-2021"),(5, 2, "Enjoy dude...", "03-01-2021"),(6, 4, "Slay
baby gurl...", "03-04-2019");

-- Inserting data in likes table.
INSERT INTO likes (post_id, user_id, like_date) VALUES (1, 2,
"03-05-2023"), (2, 4, "03-05-2023"), (2, 6, "07-11-2021"), (5, 1, "03-02-
2021"),(6, 5, "05-04-2019"),(1, 1, "13-05-2023");

--This query display the data of user, post, comment and like table.
SELECT * FROM user;
SELECT * FROM likes;
SELECT user_id, Caption FROM posts ;
SELECT comment FROM comments;

--This query display the user name as name and gender of female users.
SELECT user_name AS name, gender FROM user WHERE gender = "Female";
```

```sql
--In this query we are altering the table and adding a new column named age and setting values of age for each user.
ALTER TABLE user ADD age INTEGER;
SELECT * FROM user;
UPDATE user SET age = "21" WHERE id = 1;
UPDATE user SET age = "34" WHERE id = 2;
UPDATE user SET age = "40" WHERE id = 3;
UPDATE user SET age = "35" WHERE id = 4;
UPDATE user SET age = "38" WHERE id = 5;
UPDATE user SET age = "22" WHERE id = 6;

SELECT * FROM user;

--This query display the name of age category of users.
SELECT user_name, age,
CASE
WHEN age < 24 THEN "Students"
WHEN age <= 35 THEN "Matured"
WHEN age > 35 THEN "Adults"
END AS "age_category"
FROM user;

--This query display the average age of instagram users.
SELECT Avg(age) AS average_age_of_users FROM user;

--This query display the average age of instagram user.
SELECT user_name, Min(age) AS Minimum_age_of_user FROM user;

--This query display the average age of instagram users.
SELECT User_name, MAX(age) AS Maximum_age_of_user FROM user;

--This query arrange the data in ascending order according to age.
SELECT user_name, age AS sorted_age FROM user ORDER BY age ASC;
```

```sql
--This query count the number of users having age more than 28.
SELECT count(*) FROM user WHERE age > 28;

--This query displays the count of user group by their gender.
SELECT gender, count(*) AS total_users FROM user GROUP BY gender
HAVING COUNT(*);

--This query displays the username and caption by using subqueries.
SELECT (SELECT user_name FROM user WHERE user.id = posts.user_id) AS
user_name, caption FROM posts WHERE user_id IN(SELECT id FROM user
WHERE age = 21);

--This query display the name of user along with there post caption by
joining two tables.
SELECT user_name, caption FROM posts
JOIN user
ON user.id = posts.user_id;

--This query display the name of user along with there post caption by
joining two tables.
SELECT user_name, comment, caption FROM user
LEFT OUTER JOIN comments
ON user.id = comments.user_id
LEFT OUTER JOIN posts
ON user.id = posts.user_id;

--This query deletes all the rows from the likes table where date is 07-11-
2021.
DELETE FROM likes WHERE like_date like "07-11-2021";
SELECT user_name, like_date FROM likes
JOIN user
ON user.id = likes.user_id;
```

Conclusion :

This SQL mini-project provided a practical understanding of how data is managed in a relational database, especially in social platforms. Through the creation of tables and execution of various queries, it helped reinforce the theoretical knowledge of SQL and relational data modeling. This project is a strong foundation for future projects involving backend databases, web applications, or data analytics.