

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
A.Y. 2023 - 24
Course: Database Management Systems

Project Report

| | | |
|--------------------------------|----------------|---------------------|
| Program | MBA Tech AI | |
| Semester | 4 | |
| Name of the Project: | Connect Sphere | |
| Details of Project Members | | |
| Batch | Roll No. | Name |
| B1 | R024 | Mannarsh Singh Bali |
| Date of Submission: 03/04/2024 | | |

Contribution of each project Members:

| Roll No. | Name: | Contribution |
|----------|---------------------|---|
| R024 | Mannarsh Singh Bali | Data collection and organization Relational model Table Creation Data Insertion Attributes Tables ER diagram |

Github link of your project:

[New File at / . Mannarsh22/Connect-Sphere \(github.com\)](https://github.com/Mannarsh22/Connect-Sphere)

Project Report

CONNECT SPHERE
By

Mannarsh Singh Bali, Roll number: R024

Course: DBMS

AY: 2023-24

Table of Contents

| Sr no. | Topic | Page no. |
|--------|--|----------|
| 1 | Storyline | 1 |
| 2 | Components of Database Design | 4 |
| 3 | Entity Relationship Diagram | 6 |
| 4 | Relational Model | 7 |
| 5 | Normalization | 7 |
| 6 | SQL Queries | 9 |
| 7 | Learning from the Project | 23 |
| 8 | Challenges you faced while doing the Project | 25 |
| 9 | Conclusion | 28 |

I.

Storyline

Introduction to Connect Sphere:

Connect Sphere is an innovative Database Management System (DBMS) project tailored to revolutionize the way users interact, share, and engage on a social media platform. The project's mission is to establish a centralized platform that fosters dynamic social interactions, content sharing, and community building among users worldwide.

Core Features of Connect Sphere:

At the heart of Connect Sphere is a powerful database system meticulously designed to store comprehensive information about users, their posts, comments, likes, follows, group memberships, direct messages, notifications, and user settings. This expansive database is the foundation for Connect Sphere's ability to provide a seamless and enriching social media experience. It empowers users to create and share content effortlessly, connect with others based on shared interests, and participate in groups and discussions. Additionally, the platform includes advanced features for privacy management, content recommendations, and real-time notifications, ensuring that users have complete control over their online presence and interactions.

Mission and Vision:

By integrating these diverse aspects into a cohesive database architecture, Connect Sphere aims to enhance the digital social fabric, making it more connected, accessible, and engaging for users around the globe."

II.

Components of Database Design

Users:

Manages details of social media platform users, including their contact information and roles.

- Attributes:

| |
|-----------------------|
| userId (Primary Key), |
| Username |
| Email |
| PasswordHash |
| Full Name |
| Bio |
| Profile Pic |
| JoinDate |

Posts:

Stores posts made by users, including content, media links, and status.

- Attributes:

| |
|---------------------------------------|
| postId (Primary Key) |
| userId (Foreign Key references Users) |
| content |
| mediaUrl |
| PostDate |
| Location |

Comments:

Tracks comments made on posts, including the commenter's ID and the post ID.

- Attributes:

| |
|---------------------------------------|
| commentId (Primary Key) |
| postId (Foreign Key references Posts) |
| userId (Foreign Key references Users) |
| content |
| commentDate |

User Settings:

Captures detailed profiles of users, including personal and preference details.

- Attributes:

| |
|---------------------------------------|
| SettingId (Primary Key) |
| userId (Foreign Key references Users) |
| SettingType |
| SettingValue |

Likes:

Monitors interactions with content, like likes and shares, including post ID and user ID.

- Attributes:

| |
|---------------------------------------|
| LikeId (Primary Key) |
| UserId (Foreign Key references Users) |
| PostId (Foreign Key references Posts) |
| CommentID |
| LikeDate |

Followers:

Documents users' connections, like friends or followers.

- Attributes:

| |
|--------------------------|
| FollowerId (Primary Key) |
| FollowingIserID |
| FollowedUserID |
| FollowDate |

Groups:

Handles details about user-created or interest groups.

- Attributes:

| |
|-----------------------|
| groupId (Primary Key) |
| groupName |
| description |
| createDate |

Group Memberships:

Manages memberships of users in groups, including roles within the group.

- Attributes:

| |
|---|
| membershipId (Primary Key) |
| groupId (Foreign Key references Groups) |
| userId (Foreign Key references Users) |
| joinDate |

Messages:

Stores information about private messages between users.

- Attributes:

| |
|---|
| messageId |
| senderId (Foreign Key references Users) |
| receiverId (Foreign Key references Users) |
| messageContent |
| sendDate |

Notifications:

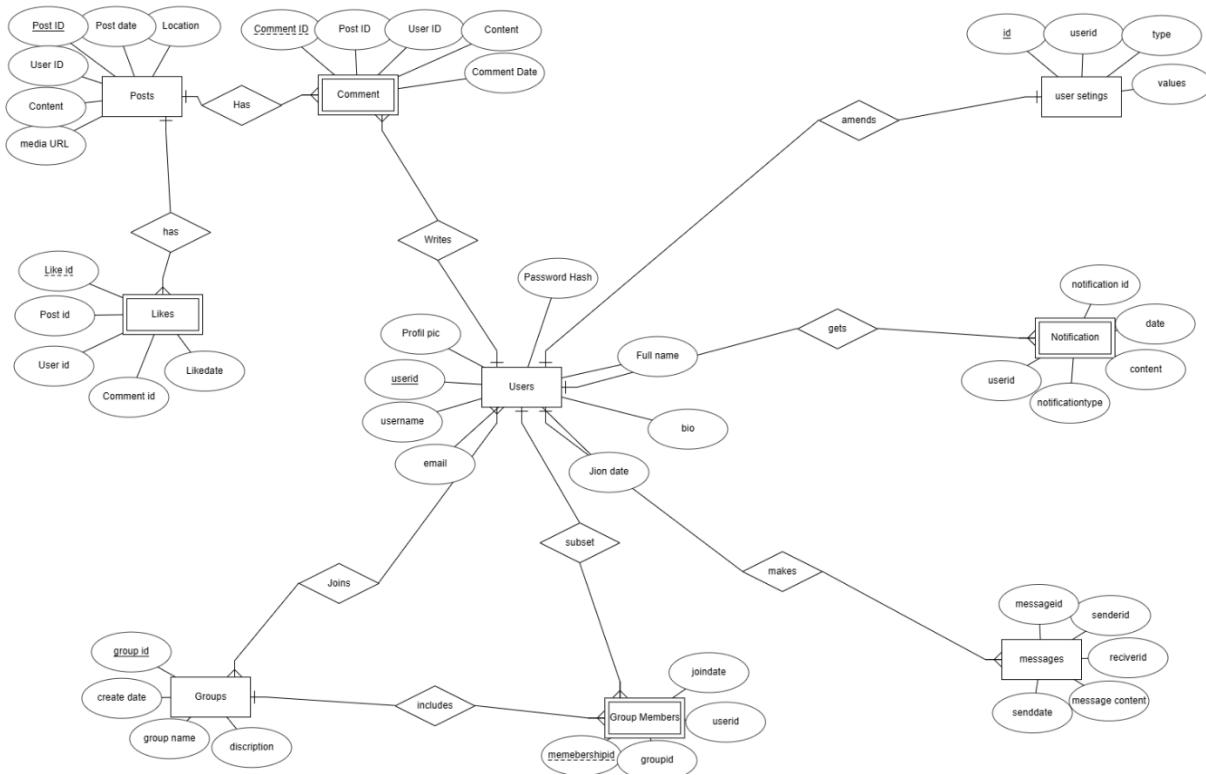
Tracks notifications sent to users, like new follows or comments.

- Attributes:

| |
|---|
| notificationId, |
| userId (Foreign Key references Users) |
| notificationType (newFollow, newComment, newLike) |
| notificationContent |
| notificationDate |

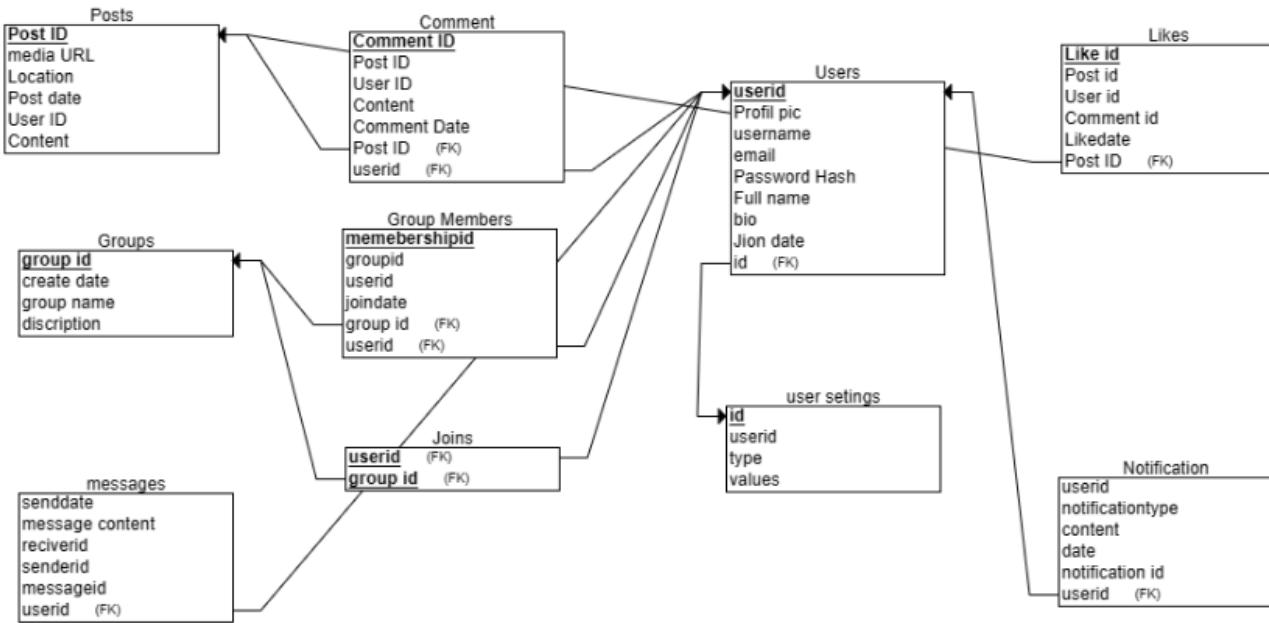
III.

Entity Relationship Diagram



IV.

Relational Model



| | |
|---|---|
| Users Create Posts | Users Make Comments |
| Users 1:M Posts Cardinality: One to Many Participation: Total (Users), Total (Posts) | Users 1:M Comments Cardinality: One to Many Participation: Total (Users), Total (Comments) |
| Posts Receive Comments | Users Interact with Content |
| Posts 1:M Comments Cardinality: One to Many Participation: Total (Posts), Total (Comments) | Users M:M Posts (via Likes and Shares) Cardinality: Many to Many Participation: Total (Users), Partial (Posts) |
| Users Follow Each Other | Users Join Groups |
| Users M:M Users (Followers and Following) Cardinality: Many to Many Participation: Total (Users), Total (Users) | Users M:M Groups (via Group Memberships) Cardinality: Many to Many Participation: Total (Users), Total (Groups) |

| | |
|---|---|
| Users Send Messages Users 1:M Direct Messages Cardinality: One to Many (for both sender and receiver) Participation: Total (Users), Total (Direct Messages) | Users Receive Notifications Users 1:M Notifications Cardinality: One to Many Participation: Total (Users), Total (Notifications) |
| Users Participate in Group Discussions Groups 1:M Posts Cardinality: One to Many Participation: Total (Groups), Partial (Posts) | Group Admins Manage Groups Users 1:M Groups (as Admins) Cardinality: One to Many Participation: Partial (Users as Admins), Total (Groups) |
| Content Tagged in Posts Posts M:M Tags (Content Tagging) Cardinality: Many to Many Participation: Total (Posts), Total (Tags) | Users Share Posts Posts M:M Posts (Shared Posts) Cardinality: Many to Many Participation: Partial (Original Posts), Total (Shared Posts) |

V.

Normalization

Atomicity: All tables have atomic columns, ensuring that each column contains only one piece of data.

No partial dependencies: There are no partial dependencies within any table, meaning that each non-prime attribute is fully functionally dependent on the primary key.

No transitive dependencies: Transitive dependencies, where a non-prime attribute depends on another non-prime attribute, are absent in all tables.

BCNF Compliance: Each table satisfies BCNF, ensuring that all functional dependencies are determined by candidate keys. This level of normalization helps to reduce redundancy, update anomalies, and ensures data integrity within the database. Additionally, foreign key constraints are appropriately defined to maintain referential integrity across tables.

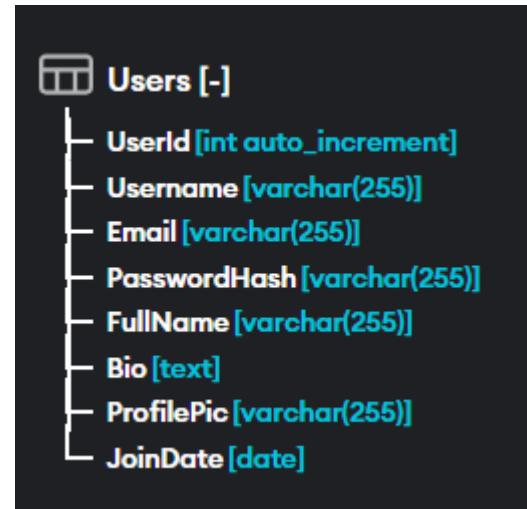
VI.

SQL Queries

TABLE CREATION CODES

Table One

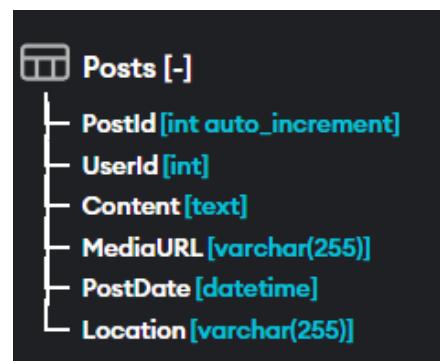
```
-- Users
CREATE TABLE Users (
    UserId INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(255) NOT NULL UNIQUE,
    Email VARCHAR(255) NOT NULL UNIQUE,
    PasswordHash VARCHAR(255) NOT NULL,
    FullName VARCHAR(255),
    Bio TEXT,
    ProfilePic VARCHAR(255),
    JoinDate DATE NOT NULL
);
```



| Users | | | | | | |
|--------|----------|-------|--------------|----------|-----|-----|
| UserId | Username | Email | PasswordHash | FullName | Bio | Pro |
| empty | | | | | | |

Table Two

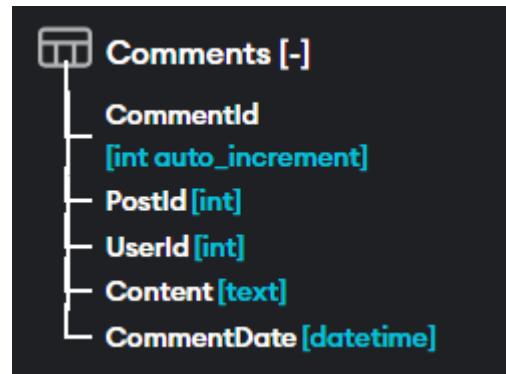
```
-- Posts
CREATE TABLE Posts (
    PostId INT AUTO_INCREMENT PRIMARY KEY,
    UserId INT,
    Content TEXT,
    MediaURL VARCHAR(255),
    PostDate DATETIME NOT NULL,
    Location VARCHAR(255),
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
);
```



| Posts | | | | | |
|--------|--------|---------|----------|----------|----------|
| PostId | UserId | Content | MediaURL | PostDate | Location |
| empty | | | | | |

Table Three

```
-- Comments
CREATE TABLE Comments (
    CommentId INT AUTO_INCREMENT PRIMARY KEY,
    PostId INT,
    UserId INT,
    Content TEXT,
    CommentDate DATETIME NOT NULL,
    FOREIGN KEY (PostId) REFERENCES Posts(PostId),
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
);
```

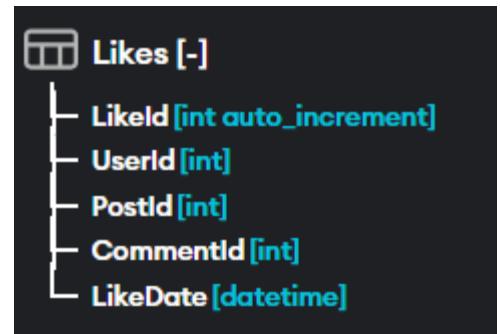


Comments

| CommentId | PostId | UserId | Content | CommentDate |
|-----------|--------|--------|---------|-------------|
| empty | | | | |

Table Four

```
-- Likes
CREATE TABLE Likes (
    Liked INT AUTO_INCREMENT PRIMARY KEY,
    UserId INT,
    PostId INT NULL,
    CommentId INT NULL,
    LikeDate DATETIME NOT NULL,
    FOREIGN KEY (UserId) REFERENCES Users(UserId),
    FOREIGN KEY (PostId) REFERENCES Posts(PostId),
    FOREIGN KEY (CommentId) REFERENCES
    Comments(CommentId)
);
```

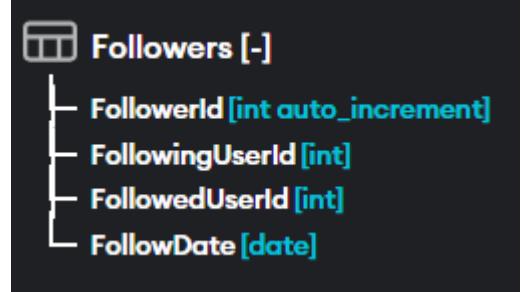


Likes

| Liked | UserId | PostId | CommentId | LikeDate |
|-------|--------|--------|-----------|----------|
| empty | | | | |

Table Five

```
-- Followers
CREATE TABLE Followers (
    FollowerId INT AUTO_INCREMENT PRIMARY KEY,
    FollowingUserId INT,
    FollowedUserId INT,
    FollowDate DATE NOT NULL,
    FOREIGN KEY (FollowingUserId) REFERENCES Users(UserId),
    FOREIGN KEY (FollowedUserId) REFERENCES Users(UserId)
);
```

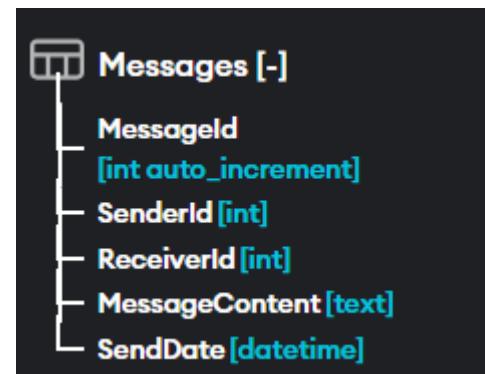


Followers

| FollowerId | FollowingUserId | FollowedUserId | FollowDate |
|------------|-----------------|----------------|------------|
| empty | | | |

Table Six

```
-- Messages
CREATE TABLE Messages (
    MessageId INT AUTO_INCREMENT PRIMARY KEY,
    SenderId INT,
    ReceiverId INT,
    MessageContent TEXT,
    SendDate DATETIME NOT NULL,
    FOREIGN KEY (SenderId) REFERENCES Users(UserId),
    FOREIGN KEY (ReceiverId) REFERENCES Users(UserId)
);
```

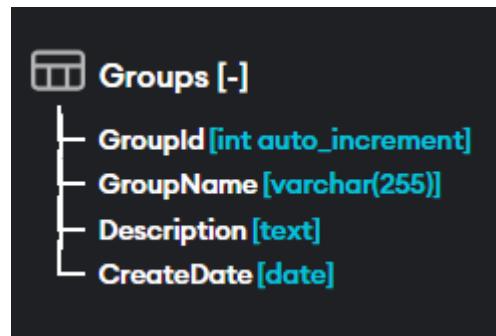


Messages

| MessageId | SenderId | ReceiverId | MessageContent | SendDate |
|-----------|----------|------------|----------------|----------|
| empty | | | | |

Table Seven

```
-- Groups
CREATE TABLE Groups (
    GroupId INT AUTO_INCREMENT PRIMARY KEY,
    GroupName VARCHAR(255) NOT NULL,
    Description TEXT,
    CreateDate DATE NOT NULL
);
```

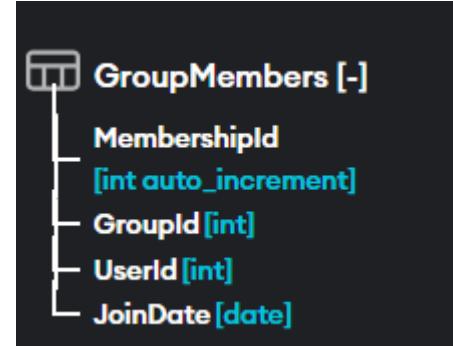


Groups

| GroupId | GroupName | Description | CreateDate |
|---------|-----------|-------------|------------|
| empty | | | |

Table Eight

```
-- GroupMembers
CREATE TABLE GroupMembers (
    MembershipId INT AUTO_INCREMENT PRIMARY KEY,
    GroupId INT,
    UserId INT,
    JoinDate DATE NOT NULL,
    FOREIGN KEY (GroupId) REFERENCES Groups(GroupId),
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
);
```

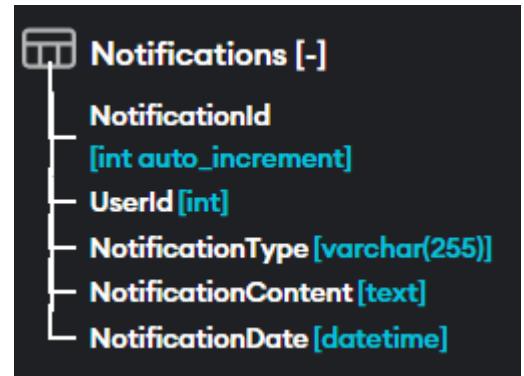


GroupMembers

| MembershipId | GroupId | UserId | JoinDate |
|--------------|---------|--------|----------|
| empty | | | |

Table Nine

```
-- Notifications
CREATE TABLE Notifications (
    NotificationId INT AUTO_INCREMENT PRIMARY KEY,
    UserId INT,
    NotificationType VARCHAR(255) NOT NULL,
    NotificationContent TEXT,
    NotificationDate DATETIME NOT NULL,
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
);
```

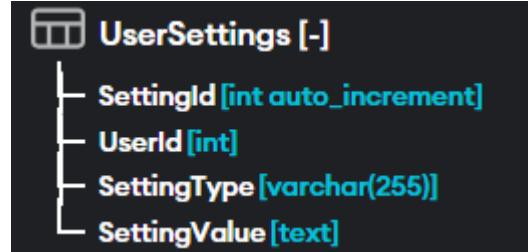


Notifications

| NotificationId | UserId | NotificationType | NotificationContent | NotificationDate |
|----------------|--------|------------------|---------------------|------------------|
| empty | | | | |

Table Ten

```
-- UserSettings
CREATE TABLE UserSettings (
    SettingId INT AUTO_INCREMENT PRIMARY KEY,
    UserId INT,
    SettingType VARCHAR(255) NOT NULL,
    SettingValue TEXT NOT NULL,
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
);
```



UserSettings

| SettingId | UserId | SettingType | SettingValue |
|-----------|--------|-------------|--------------|
| empty | | | |

INSERTION CODES

Table one

```
INSERT INTO Users (Username, Email, PasswordHash, FullName, Bio, ProfilePic, JoinDate) VALUES
('johnDoe', 'john.doe@email.com', 'hashed_password_1', 'John Doe', 'Just a regular Joe.', 'profilepic1.jpg', '2023-01-01'),
('janeDoe', 'jane.doe@email.com', 'hashed_password_2', 'Jane Doe', 'Love hiking and outdoor adventures.', 'profilepic2.jpg', '2023-01-02'),
('coolUser123', 'cooluser123@email.com', 'hashed_password_3', 'Cool User', 'Always up for a cool adventure.', 'profilepic3.jpg', '2023-01-03'),
('travelBug', 'travel.bug@email.com', 'hashed_password_4', 'Travel Bug', 'Exploring the world, one country at a time.', 'profilepic4.jpg', '2023-01-04'),
('bookworm', 'bookworm@email.com', 'hashed_password_5', 'Book Worm', 'A room without books is like a body without a soul.', 'profilepic5.jpg', '2023-01-05'),
('gamerGuy', 'gamer.guy@email.com', 'hashed_password_6', 'Gamer Guy', 'Living one game at a time.', 'profilepic6.jpg', '2023-01-06'),
('codingNinja', 'coding.ninja@email.com', 'hashed_password_7', 'Coding Ninja', 'Writing code and fixing bugs.', 'profilepic7.jpg', '2023-01-07'),
('fitnessFreak', 'fitness.freak@email.com', 'hashed_password_8', 'Fitness Freak', 'Sweat now, shine later.', 'profilepic8.jpg', '2023-01-08'),
('foodie', 'foodie@email.com', 'hashed_password_9', 'Foodie', 'Life is short, eat dessert first.', 'profilepic9.jpg', '2023-01-09'),
('movieBuff', 'movie.buff@email.com', 'hashed_password_10', 'Movie Buff', 'A movie a day keeps reality away.', 'profilepic10.jpg', '2023-01-10'),
('techGeek', 'tech.geek@email.com', 'hashed_password_11', 'Tech Geek', 'Embracing technology, one gadget at a time.', 'profilepic11.jpg', '2023-01-11'),
('natureLover', 'nature.lover@email.com', 'hashed_password_12', 'Nature Lover', 'Finding peace in the heart of nature.', 'profilepic12.jpg', '2023-01-12'),
('musicMaestro', 'music.maestro@email.com', 'hashed_password_13', 'Music Maestro', 'Life is a song, love is the music.', 'profilepic13.jpg', '2023-01-13'),
('artAficionado', 'art.aficionado@email.com', 'hashed_password_14', 'Art Aficionado', 'Art is not what you see, but what you make others see.', 'profilepic14.jpg', '2023-01-14'),
('scienceSavvy', 'science.savvy@email.com', 'hashed_password_15', 'Science Savvy', 'Curiosity has its own reason for existing.', 'profilepic15.jpg', '2023-01-15');
```

| Users | | | | | | | |
|---------------|--------------------------|-------|--------------------|----------------|--|------------------|-------|
| UserId | Username | Email | PasswordHash | FullName | Bio | ProfilePic | Join |
| johnDoe | john.doe@email.com | | hashed_password_1 | John Doe | Just a regular Joe. | profilepic1.jpg | 20201 |
| janeDoe | jane.doe@email.com | | hashed_password_2 | Jane Doe | Love hiking and outdoor adventures. | profilepic2.jpg | 20202 |
| coolUser123 | cooluser123@email.com | | hashed_password_3 | Cool User | Always up for a cool adventure. | profilepic3.jpg | 20203 |
| travelBug | travel.bug@email.com | | hashed_password_4 | Travel Bug | Exploring the world, one country at a time. | profilepic4.jpg | 20204 |
| bookworm | bookworm@email.com | | hashed_password_5 | Book Worm | A room without books is like a body without a soul. | profilepic5.jpg | 20205 |
| gamerGuy | gamer.guy@email.com | | hashed_password_6 | Gamer Guy | Living one game at a time. | profilepic6.jpg | 20206 |
| codingNinja | coding.ninja@email.com | | hashed_password_7 | Coding Ninja | Writing code and fixing bugs. | profilepic7.jpg | 20207 |
| fitnessFreak | fitness.freak@email.com | | hashed_password_8 | Fitness Freak | Sweat now, shine later. | profilepic8.jpg | 20208 |
| foodie | foodie@email.com | | hashed_password_9 | Foodie | Life is short, eat dessert first. | profilepic9.jpg | 20209 |
| movieBuff | movie.buff@email.com | | hashed_password_10 | Movie Buff | A movie a day keeps reality away. | profilepic10.jpg | 20210 |
| techGeek | tech.geek@email.com | | hashed_password_11 | Tech Geek | Embracing technology, one gadget at a time. | profilepic11.jpg | 20211 |
| natureLover | nature.lover@email.com | | hashed_password_12 | Nature Lover | Finding peace in the heart of nature. | profilepic12.jpg | 20212 |
| musicMaestro | music.maestro@email.com | | hashed_password_13 | Music Maestro | Life is a song, love is the music. | profilepic13.jpg | 20213 |
| artAficionado | art.aficionado@email.com | | hashed_password_14 | Art Aficionado | Art is not what you see, but what you make others see. | profilepic14.jpg | 20214 |
| | | | | | Curiosity | | |

Table Two

```
INSERT INTO Posts (UserId, Content, MediaURL, PostDate, Location) VALUES
(1, 'Loving the new cafe in town!', "", '2023-01-02 10:30:00', 'Town Center'),
(2, 'Here's to new adventures!', 'adventure.jpg', '2023-01-03 09:15:00', ""),
(3, 'Never too old to go on adventures.', 'mountain.jpg', '2023-01-04 08:45:00', 'Rocky Mountains'),
(4, 'My latest read. Highly recommend!', 'book.jpg', '2023-01-05 17:20:00', ""),
(5, 'Game night was epic!', 'gamenight.jpg', '2023-01-06 20:00:00', ""),
(6, 'Coding my way through life.', "", '2023-01-07 13:00:00', 'Coffee Shop'),
(7, 'Morning run completed!', 'morningrun.jpg', '2023-01-08 07:30:00', 'Park'),
(8, 'Who else loves chocolate cake? 😊', 'chocolatecake.jpg', '2023-01-09 19:00:00', ""),
(9, 'Movie marathon, anyone?', "", '2023-01-10 21:00:00', 'Home Cinema'),
(10, 'Exploring the latest tech at the expo.', 'techexpo.jpg', '2023-01-11 15:00:00', 'Tech Expo'),
(11, 'A peaceful day in the forest.', 'forest.jpg', '2023-01-12 11:00:00', 'Greenwood Forest'),
(12, 'Jamming to some new tunes.', 'music.jpg', '2023-01-13 18:30:00', ""),
(13, 'Visited an amazing art gallery today.', 'artgallery.jpg', '2023-01-14 14:00:00', 'Downtown'),
(14, 'Stargazing night. The universe is fascinating!', 'stars.jpg', '2023-01-15 22:00:00', 'Hilltop'),
(15, 'Experimenting with new science projects.', 'scienceproject.jpg', '2023-01-16 16:00:00', 'Lab');
```

| Posts | | | | | |
|--------|--------|--|--------------------|---------------------|------------------|
| PostId | UserId | Content | MediaURL | PostDate | Location |
| 1 | 1 | Loving the new cafe in town! | | 2023-01-02 10:30:00 | Town Center |
| 2 | 2 | Here's to new adventures! | adventure.jpg | 2023-01-03 09:15:00 | |
| 3 | 3 | Never too old to go on adventures. | mountain.jpg | 2023-01-04 08:45:00 | Rocky Mountains |
| 4 | 4 | My latest read. Highly recommend! | book.jpg | 2023-01-05 17:20:00 | |
| 5 | 5 | Game night was epic! | gamenight.jpg | 2023-01-06 20:00:00 | |
| 6 | 6 | Coding my way through life. | | 2023-01-07 13:00:00 | Coffee Shop |
| 7 | 7 | Morning run completed! | morningrun.jpg | 2023-01-08 07:30:00 | Park |
| 8 | 8 | Who else loves chocolate cake? 😊 | chocolatecake.jpg | 2023-01-09 19:00:00 | |
| 9 | 9 | Movie marathon, anyone? | | 2023-01-10 21:00:00 | Home Cinema |
| 10 | 10 | Exploring the latest tech at the expo. | techexpo.jpg | 2023-01-11 15:00:00 | Tech Expo |
| 11 | 11 | A peaceful day in the forest. | forest.jpg | 2023-01-12 11:00:00 | Greenwood Forest |
| 12 | 12 | Jamming to some new tunes. | music.jpg | 2023-01-13 18:30:00 | |
| 13 | 13 | Visited an amazing art gallery today. | artgallery.jpg | 2023-01-14 14:00:00 | Downtown |
| 14 | 14 | Stargazing night. The universe is fascinating! | stars.jpg | 2023-01-15 22:00:00 | Hilltop |
| 15 | 15 | Experimenting with new science projects. | scienceproject.jpg | 2023-01-16 16:00:00 | Lab |

Table Three

```
INSERT INTO Comments (PostId, UserId, Content, CommentDate) VALUES
(1, 2, 'Absolutely love that place!', '2023-01-02 11:00:00'),
(2, 1, 'Looks amazing! Have fun!', '2023-01-03 10:00:00'),
(3, 4, 'Wish I was there with you.', '2023-01-04 09:30:00'),
(4, 3, 'That book is on my list too!', '2023-01-05 18:00:00'),
(5, 6, 'Next time count me in!', '2023-01-06 21:05:00'),
(6, 5, 'How do you find time to code?', '2023-01-07 14:00:00'),
(7, 8, 'Running is not for me, but good job!', '2023-01-08 08:00:00'),
(8, 7, 'I could eat an entire cake myself.', '2023-01-09 20:00:00'),
(9, 10, 'I'm definitely up for a movie night.', '2023-01-10 22:00:00'),
(10, 9, 'What was your favorite gadget?', '2023-01-11 16:00:00'),
(11, 12, 'Nature is so healing. Beautiful pic!', '2023-01-12 12:00:00'),
(12, 11, 'What are you listening to?', '2023-01-13 19:00:00'),
(13, 14, 'Art feeds the soul. Looks like a great day!', '2023-01-14 15:00:00'),
(14, 13, 'The stars are just magical.', '2023-01-15 23:00:00'),
(15, 1, 'Science for the win!', '2023-01-16 17:00:00');
```

| Comments | | | | |
|-----------|--------|--------|---|---------------------|
| CommentId | PostId | UserId | Content | CommentDate |
| 1 | 1 | 2 | Absolutely love that place! | 2023-01-02 11:00:00 |
| 2 | 2 | 1 | Looks amazing! Have fun! | 2023-01-03 10:00:00 |
| 3 | 3 | 4 | Wish I was there with you. | 2023-01-04 09:30:00 |
| 4 | 4 | 3 | That book is on my list too! | 2023-01-05 18:00:00 |
| 5 | 5 | 6 | Next time count me in! | 2023-01-06 21:05:00 |
| 6 | 6 | 5 | How do you find time to code? | 2023-01-07 14:00:00 |
| 7 | 7 | 8 | Running is not for me, but good job! | 2023-01-08 08:00:00 |
| 8 | 8 | 7 | I could eat an entire cake myself. | 2023-01-09 20:00:00 |
| 9 | 9 | 10 | I'm definitely up for a movie night. | 2023-01-10 22:00:00 |
| 10 | 10 | 9 | What was your favorite gadget? | 2023-01-11 16:00:00 |
| 11 | 11 | 12 | Nature is so healing. Beautiful pic! | 2023-01-12 12:00:00 |
| 12 | 12 | 11 | What are you listening to? | 2023-01-13 19:00:00 |
| 13 | 13 | 14 | Art feeds the soul. Looks like a great day! | 2023-01-14 15:00:00 |
| 14 | 14 | 13 | The stars are just magical. | 2023-01-15 23:00:00 |
| 15 | 15 | 1 | Science for the win! | 2023-01-16 17:00:00 |

Table Four

```
INSERT INTO Likes (UserId, PostId, CommentId, LikeDate) VALUES  
(2, 1, NULL, '2023-01-02 12:00:00'),  
(3, NULL, 1, '2023-01-02 12:05:00'),  
(4, 2, NULL, '2023-01-03 10:30:00'),  
(5, NULL, 2, '2023-01-03 11:00:00'),  
(1, 3, NULL, '2023-01-04 10:00:00'),  
(6, NULL, 3, '2023-01-04 10:05:00'),  
(7, 4, NULL, '2023-01-05 18:30:00'),  
(8, NULL, 4, '2023-01-05 19:00:00'),  
(9, 5, NULL, '2023-01-06 21:10:00'),  
(10, NULL, 5, '2023-01-06 21:15:00'),  
(11, 6, NULL, '2023-01-07 14:30:00'),  
(12, NULL, 6, '2023-01-07 14:35:00'),  
(13, 7, NULL, '2023-01-08 08:30:00'),  
(14, NULL, 7, '2023-01-08 08:35:00'),  
(15, 8, NULL, '2023-01-09 20:30:00');
```

| Likes | | | | |
|-------|--------|--------|-----------|---------------------|
| LikId | UserId | PostId | CommentId | LikeDate |
| 1 | 2 | 1 | | 2023-01-02 12:00:00 |
| 2 | 3 | | 1 | 2023-01-02 12:05:00 |
| 3 | 4 | 2 | | 2023-01-03 10:30:00 |
| 4 | 5 | | 2 | 2023-01-03 11:00:00 |
| 5 | 1 | 3 | | 2023-01-04 10:00:00 |
| 6 | 6 | | 3 | 2023-01-04 10:05:00 |
| 7 | 7 | 4 | | 2023-01-05 18:30:00 |
| 8 | 8 | | 4 | 2023-01-05 19:00:00 |
| 9 | 9 | 5 | | 2023-01-06 21:10:00 |
| 10 | 10 | | 5 | 2023-01-06 21:15:00 |
| 11 | 11 | 6 | | 2023-01-07 14:30:00 |
| 12 | 12 | | 6 | 2023-01-07 14:35:00 |
| 13 | 13 | 7 | | 2023-01-08 08:30:00 |
| 14 | 14 | | 7 | 2023-01-08 08:35:00 |
| 15 | 15 | 8 | | 2023-01-09 20:30:00 |

Table Five

```
INSERT INTO Followers (FollowingUserId, FollowedUserId, FollowDate) VALUES  
(1, 2, '2023-01-01'),  
(1, 3, '2023-01-02'),  
(2, 1, '2023-01-03'),  
(3, 4, '2023-01-04'),  
(4, 5, '2023-01-05'),  
(5, 6, '2023-01-06'),  
(6, 7, '2023-01-07'),  
(7, 8, '2023-01-08'),  
(8, 9, '2023-01-09'),  
(9, 10, '2023-01-10'),  
(10, 11, '2023-01-11'),  
(11, 12, '2023-01-12'),  
(12, 13, '2023-01-13'),  
(13, 14, '2023-01-14'),  
(14, 15, '2023-01-15');
```

| Followers | | | |
|-----------|-----------------|----------------|------------|
| FollowId | FollowingUserId | FollowedUserId | FollowDate |
| 1 | 1 | 2 | 2023-01-01 |
| 2 | 1 | 3 | 2023-01-02 |
| 3 | 2 | 1 | 2023-01-03 |
| 4 | 3 | 4 | 2023-01-04 |
| 5 | 4 | 5 | 2023-01-05 |
| 6 | 5 | 6 | 2023-01-06 |
| 7 | 6 | 7 | 2023-01-07 |
| 8 | 7 | 8 | 2023-01-08 |
| 9 | 8 | 9 | 2023-01-09 |
| 10 | 9 | 10 | 2023-01-10 |
| 11 | 10 | 11 | 2023-01-11 |
| 12 | 11 | 12 | 2023-01-12 |
| 13 | 12 | 13 | 2023-01-13 |
| 14 | 13 | 14 | 2023-01-14 |
| 15 | 14 | 15 | 2023-01-15 |

Table Six

```
INSERT INTO Messages (SenderId, ReceiverId, MessageContent, SendDate) VALUES
(1, 2, 'Hey, how have you been?', '2023-01-02 10:00:00'),
(2, 1, 'I'm good, thanks! And you?', '2023-01-02 10:05:00'),
(3, 4, 'Are we still on for tomorrow?', '2023-01-03 15:30:00'),
(4, 3, 'Yes, looking forward to it!', '2023-01-03 15:45:00'),
(5, 6, 'Did you see that new movie?', '2023-01-04 20:00:00'),
(6, 5, 'Not yet, is it any good?', '2023-01-04 20:10:00'),
(7, 8, 'Happy Birthday!! 🎉', '2023-01-05 00:01:00'),
(8, 7, 'Thank you so much!! 😊', '2023-01-05 08:30:00'),
(9, 10, 'Wanna grab lunch this week?', '2023-01-06 12:00:00'),
(10, 9, 'Sure, how about Wednesday?', '2023-01-06 12:15:00'),
(11, 12, 'Can you help me with my project?', '2023-01-07 17:00:00'),
(12, 11, 'Of course, what do you need?', '2023-01-07 17:05:00'),
(13, 14, 'I loved the photos you posted!', '2023-01-08 09:00:00'),
(14, 13, 'Thanks! Glad you liked them 😊', '2023-01-08 09:10:00'),
(15, 1, 'We should catch up sometime!', '2023-01-09 11:00:00');
```

| Messages | | | | |
|-----------|----------|------------|----------------------------------|---------------------|
| MessageId | SenderId | ReceiverId | MessageContent | SendDate |
| 1 | 1 | 2 | Hey, how have you been? | 2023-01-02 10:00:00 |
| 2 | 2 | 1 | I'm good, thanks! And you? | 2023-01-02 10:05:00 |
| 3 | 3 | 4 | Are we still on for tomorrow? | 2023-01-03 15:30:00 |
| 4 | 4 | 3 | Yes, looking forward to it! | 2023-01-03 15:45:00 |
| 5 | 5 | 6 | Did you see that new movie? | 2023-01-04 20:00:00 |
| 6 | 6 | 5 | Not yet, is it any good? | 2023-01-04 20:10:00 |
| 7 | 7 | 8 | Happy Birthday!! 🎉 | 2023-01-05 00:01:00 |
| 8 | 8 | 7 | Thank you so much!! 😊 | 2023-01-05 08:30:00 |
| 9 | 9 | 10 | Wanna grab lunch this week? | 2023-01-06 12:00:00 |
| 10 | 10 | 9 | Sure, how about Wednesday? | 2023-01-06 12:15:00 |
| 11 | 11 | 12 | Can you help me with my project? | 2023-01-07 17:00:00 |
| 12 | 12 | 11 | Of course, what do you need? | 2023-01-07 17:05:00 |
| 13 | 13 | 14 | I loved the photos you posted! | 2023-01-08 09:00:00 |
| 14 | 14 | 13 | Thanks! Glad you liked them 😊 | 2023-01-08 09:10:00 |
| 15 | 15 | 1 | We should catch up sometime! | 2023-01-09 11:00:00 |

Table Seven

```
INSERT INTO Groups (GroupId, GroupName, Description, CreateDate) VALUES
(1, 'Outdoor Enthusiasts', 'A group for people who love outdoor activities', '2023-01-01'),
(2, 'Book Club', 'Discussing the latest and classic books', '2023-01-02'),
(3, 'Movie Lovers', 'A place to share and discover movies', '2023-01-03'),
(4, 'Tech Innovators', 'Exploring the latest in technology', '2023-01-04'),
(5, 'Fitness Freaks', 'Sharing fitness tips and motivation', '2023-01-05'),
(6, 'Gourmet Group', 'For those who love cooking and eating', '2023-01-06'),
(7, 'Music Makers', 'Connecting musicians and music lovers', '2023-01-07'),
(8, 'Art Admirers', 'Showcase and discuss art', '2023-01-08'),
(9, 'Science Enthusiasts', 'A hub for science lovers to share knowledge', '2023-01-09'),
(10, 'Travel Buddies', 'Share travel stories and tips', '2023-01-10'),
(11, 'Coding Corner', 'A place for developers to talk code', '2023-01-11'),
(12, 'Pet Pals', 'A group for pet owners to share their experiences', '2023-01-12'),
(13, 'Green Thumbs', 'Discussing all things gardening', '2023-01-13'),
(14, 'History Buffs', 'A group for those who love history', '2023-01-14'),
(15, 'Fashion Forward', 'For those who want to share style tips and trends', '2023-01-15');
```

| Groups | | | |
|---------|---------------------|---|------------|
| GroupId | GroupName | Description | CreateDate |
| 1 | Outdoor Enthusiasts | A group for people who love outdoor activities | 2023-01-01 |
| 2 | Book Club | Discussing the latest and classic books | 2023-01-02 |
| 3 | Movie Lovers | A place to share and discover movies | 2023-01-03 |
| 4 | Tech Innovators | Exploring the latest in technology | 2023-01-04 |
| 5 | Fitness Freaks | Sharing fitness tips and motivation | 2023-01-05 |
| 6 | Gourmet Group | For those who love cooking and eating | 2023-01-06 |
| 7 | Music Makers | Connecting musicians and music lovers | 2023-01-07 |
| 8 | Art Admirers | Showcase and discuss art | 2023-01-08 |
| 9 | Science Enthusiasts | A hub for science lovers to share knowledge | 2023-01-09 |
| 10 | Travel Buddies | Share travel stories and tips | 2023-01-10 |
| 11 | Coding Corner | A place for developers to talk code | 2023-01-11 |
| 12 | Pet Pals | A group for pet owners to share their experiences | 2023-01-12 |
| 13 | Green Thumbs | Discussing all things gardening | 2023-01-13 |
| 14 | History Buffs | A group for those who love history | 2023-01-14 |
| 15 | Fashion Forward | For those who want to share style tips and trends | 2023-01-15 |

Table Eight

```
INSERT INTO GroupMembers (MembershipId, GroupId, UserId, JoinDate) VALUES  
(1, 1, 1, '2023-01-02'),  
(2, 1, 2, '2023-01-03'),  
(3, 2, 3, '2023-01-04'),  
(4, 2, 4, '2023-01-05'),  
(5, 3, 5, '2023-01-06'),  
(6, 3, 6, '2023-01-07'),  
(7, 4, 7, '2023-01-08'),  
(8, 4, 8, '2023-01-09'),  
(9, 5, 9, '2023-01-10'),  
(10, 5, 10, '2023-01-11'),  
(11, 6, 11, '2023-01-12'),  
(12, 6, 12, '2023-01-13'),  
(13, 7, 13, '2023-01-14'),  
(14, 7, 14, '2023-01-15'),  
(15, 8, 15, '2023-01-16');
```

| GroupMembers | | | |
|--------------|---------|--------|------------|
| MembershipId | GroupId | UserId | JoinDate |
| 1 | 1 | 1 | 2023-01-02 |
| 2 | 1 | 2 | 2023-01-03 |
| 3 | 2 | 3 | 2023-01-04 |
| 4 | 2 | 4 | 2023-01-05 |
| 5 | 3 | 5 | 2023-01-06 |
| 6 | 3 | 6 | 2023-01-07 |
| 7 | 4 | 7 | 2023-01-08 |
| 8 | 4 | 8 | 2023-01-09 |
| 9 | 5 | 9 | 2023-01-10 |
| 10 | 5 | 10 | 2023-01-11 |
| 11 | 6 | 11 | 2023-01-12 |
| 12 | 6 | 12 | 2023-01-13 |
| 13 | 7 | 13 | 2023-01-14 |
| 14 | 7 | 14 | 2023-01-15 |
| 15 | 8 | 15 | 2023-01-16 |

Table Nine

```
INSERT INTO Notifications (NotificationId, UserId, NotificationType, NotificationContent, NotificationDate) VALUES
(1, 1, 'Follow', 'User 2 started following you', '2023-01-02 09:00:00'),
(2, 2, 'Like', 'User 3 liked your post', '2023-01-03 10:00:00'),
(3, 3, 'Comment', 'User 4 commented on your post', '2023-01-04 11:00:00'),
(4, 4, 'Message', 'You have a new message from User 5', '2023-01-05 12:00:00'),
(5, 5, 'GroupJoin', 'User 6 joined your group', '2023-01-06 13:00:00'),
(6, 6, 'Follow', 'User 7 started following you', '2023-01-07 14:00:00'),
(7, 7, 'Like', 'User 8 liked your comment', '2023-01-08 15:00:00'),
(8, 8, 'Comment', 'User 9 commented on your post', '2023-01-09 16:00:00'),
(9, 9, 'Message', 'You have a new message from User 10', '2023-01-10 17:00:00'),
(10, 10, 'GroupJoin', 'User 11 joined your group', '2023-01-11 18:00:00'),
(11, 11, 'Follow', 'User 12 started following you', '2023-01-12 19:00:00'),
(12, 12, 'Like', 'User 13 liked your post', '2023-01-13 20:00:00'),
(13, 13, 'Comment', 'User 14 commented on your post', '2023-01-14 21:00:00'),
(14, 14, 'Message', 'You have a new message from User 15', '2023-01-15 22:00:00'),
(15, 15, 'GroupJoin', 'User 1 joined your group', '2023-01-16 23:00:00');
```

| Notifications | | | | |
|----------------|--------|------------------|-------------------------------------|---------------------|
| NotificationId | UserId | NotificationType | NotificationContent | NotificationDate |
| 1 | 1 | Follow | User 2 started following you | 2023-01-02 09:00:00 |
| 2 | 2 | Like | User 3 liked your post | 2023-01-03 10:00:00 |
| 3 | 3 | Comment | User 4 commented on your post | 2023-01-04 11:00:00 |
| 4 | 4 | Message | You have a new message from User 5 | 2023-01-05 12:00:00 |
| 5 | 5 | GroupJoin | User 6 joined your group | 2023-01-06 13:00:00 |
| 6 | 6 | Follow | User 7 started following you | 2023-01-07 14:00:00 |
| 7 | 7 | Like | User 8 liked your comment | 2023-01-08 15:00:00 |
| 8 | 8 | Comment | User 9 commented on your post | 2023-01-09 16:00:00 |
| 9 | 9 | Message | You have a new message from User 10 | 2023-01-10 17:00:00 |
| 10 | 10 | GroupJoin | User 11 joined your group | 2023-01-11 18:00:00 |
| 11 | 11 | Follow | User 12 started following you | 2023-01-12 19:00:00 |
| 12 | 12 | Like | User 13 liked your post | 2023-01-13 20:00:00 |
| 13 | 13 | Comment | User 14 commented on your post | 2023-01-14 21:00:00 |
| 14 | 14 | Message | You have a new message from User 15 | 2023-01-15 22:00:00 |
| 15 | 15 | GroupJoin | User 1 joined your group | 2023-01-16 23:00:00 |

Table Ten

```
INSERT INTO UserSettings (SettingId, UserId, SettingType, SettingValue) VALUES
(1, 1, 'Privacy', 'Public,' ),
(2, 2, 'Privacy', 'FriendsOnly'),
(3, 3, 'Notifications', 'Enabled'),
(4, 4, 'Notifications', 'Enabled'),
(5, 5, 'Language', 'English'),
(6, 6, 'Language', 'Spanish'),
(7, 7, 'Privacy', 'Public'),
(8, 8, 'Notifications', 'Disabled'),
(9, 9, 'Language', 'French'),
(10, 10, 'Privacy', 'FriendsOnly'),
(11, 11, 'Notifications', 'Enabled'),
(12, 12, 'Language', 'German'),
(13, 13, 'Privacy', 'Public'),
(14, 14, 'Notifications', 'Disabled'),
(15, 15, 'Language', 'Japanese');
```

| UserSettings | | | |
|--------------|--------|---------------|--------------|
| SettingId | UserId | SettingType | SettingValue |
| 1 | 1 | Privacy | Public, |
| 2 | 2 | Privacy | FriendsOnly |
| 3 | 3 | Notifications | Enabled |
| 4 | 4 | Notifications | Enabled |
| 5 | 5 | Language | English |
| 6 | 6 | Language | Spanish |
| 7 | 7 | Privacy | Public |
| 8 | 8 | Notifications | Disabled |
| 9 | 9 | Language | French |
| 10 | 10 | Privacy | FriendsOnly |
| 11 | 11 | Notifications | Enabled |
| 12 | 12 | Language | German |
| 13 | 13 | Privacy | Public |
| 14 | 14 | Notifications | Disabled |
| 15 | 15 | Language | Japanese |

SQL QUERY

Query 1: Retrieve all users.

Question: How can we list all the users in our social media platform?

```
SELECT * FROM Users;
```

Output

| UserId | Username | Email | PasswordHash | FullName | Bio | ProfilePic | JoinDate |
|--------|--------------|-------------------------|-------------------|---------------|---|-----------------|------------|
| 1 | johnDoe | john.doe@email.com | hashed_password_1 | John Doe | Just a regular Joe. | profilepic1.jpg | 2023-01-01 |
| 2 | janeDoe | jane.doe@email.com | hashed_password_2 | Jane Doe | Love hiking and outdoor adventures. | profilepic2.jpg | 2023-01-02 |
| 3 | coolUser123 | cooluser123@email.com | hashed_password_3 | Cool User | Always up for a cool adventure. | profilepic3.jpg | 2023-01-03 |
| 4 | travelBug | travel.bug@email.com | hashed_password_4 | Travel Bug | Exploring the world, one country at a time. | profilepic4.jpg | 2023-01-04 |
| 5 | bookworm | bookworm@email.com | hashed_password_5 | Book Worm | A room without books is like a body without a soul. | profilepic5.jpg | 2023-01-05 |
| 6 | gamerGuy | gamer.guy@email.com | hashed_password_6 | Gamer Guy | Living one game at a time. | profilepic6.jpg | 2023-01-06 |
| 7 | codingNinja | coding.ninja@email.com | hashed_password_7 | Coding Ninja | Writing code and fixing bugs. | profilepic7.jpg | 2023-01-07 |
| 8 | fitnessFreak | fitness.freak@email.com | hashed_password_8 | Fitness Freak | Sweat now, shine later. | profilepic8.jpg | 2023-01-08 |
| 9 | foodie | foodie@email.com | hashed_password_9 | Foodie | Life is short, eat | profilepic9.jpg | 2023-01-09 |

Query 2: Find all posts made by a specific user.

Question: How do we find all posts made by the user with UserId = 1?

```
SELECT * FROM Posts WHERE UserId = 1;
```

Output

| PostId | UserId | Content | MediaURL | PostDate | Location |
|--------|--------|------------------------------|----------|---------------------|-------------|
| 1 | 1 | Loving the new cafe in town! | | 2023-01-02 10:30:00 | Town Center |

Query 3: Count the number of posts each user has made.

Question: How can we count the total number of posts made by each user?

```
SELECT UserId, COUNT(*) AS NumberOfPosts FROM Posts GROUP BY UserId;
```

| Output | |
|--------|---------------|
| UserId | NumberOfPosts |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |
| 11 | 1 |
| 12 | 1 |
| 13 | 1 |
| 14 | 1 |
| 15 | 1 |

Query 4: List users who have not made any posts.

Question: How do we find users who haven't made any posts yet?

```
SELECT * FROM Users WHERE UserId NOT IN (SELECT DISTINCT UserId FROM Posts);
```

| Output | |
|--|--|
| SQL query successfully executed. However, the result set is empty. | |

Query 5: Find the most recent post of each user.

Question: How can we retrieve the most recent post made by each user?

```
SELECT p.UserId, MAX(p.PostDate) as MostRecentPost  
FROM Posts p  
GROUP BY p.UserId;
```

| Output | |
|--------|---------------------|
| Userid | MostRecentPost |
| 1 | 2023-01-02 10:30:00 |
| 2 | 2023-01-03 09:15:00 |
| 3 | 2023-01-04 08:45:00 |
| 4 | 2023-01-05 17:20:00 |
| 5 | 2023-01-06 20:00:00 |
| 6 | 2023-01-07 13:00:00 |
| 7 | 2023-01-08 07:30:00 |
| 8 | 2023-01-09 19:00:00 |
| 9 | 2023-01-10 21:00:00 |
| 10 | 2023-01-11 15:00:00 |
| 11 | 2023-01-12 11:00:00 |
| 12 | 2023-01-13 18:30:00 |
| 13 | 2023-01-14 14:00:00 |
| 14 | 2023-01-15 22:00:00 |
| 15 | 2023-01-16 16:00:00 |

Query 6: Display all comments on a specific post.

Question: How do we show all comments for the post with PostId = 2?

```
SELECT * FROM Comments WHERE PostId = 2;
```

| Output | | | | |
|-----------|--------|--------|--------------------------|---------------------|
| CommentId | PostId | Userid | Content | CommentDate |
| 2 | 2 | 1 | Looks amazing! Have fun! | 2023-01-03 10:00:00 |

Query 7: List all likes on posts made by a specific user.

Question: How can we list all likes for posts made by the user with UserId = 1?

```
SELECT l.* FROM Likes l  
JOIN Posts p ON l.PostId = p.PostId  
WHERE p.UserId = 1;
```

| Output | | | | |
|--------|--------|--------|-----------|---------------------|
| LikId | UserId | PostId | CommentId | LikeDate |
| 1 | 2 | 1 | | 2023-01-02 12:00:00 |

Query 8: Find the total number of likes for each post.

Question: How can we count the total number of likes on each post?

```
SELECT PostId, COUNT(*) AS TotalLikes FROM Likes GROUP BY PostId;
```

| Output | |
|--------|------------|
| PostId | TotalLikes |
| | 7 |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |

Query 9: Retrieve user details along with the number of followers they have.

Question: How do we find each user's details and their follower count?

```
SELECT u.*, COUNT(f.FollowingUserId) AS NumberOfFollowers
FROM Users u
LEFT JOIN Followers f ON u.UserId = f.FollowedUserId
GROUP BY u.UserId;
```

| Output | | | | | | | | |
|--------|-------------|-----------------------|-------------------|------------|---|-----------------|------------|-----------------|
| UserId | Username | Email | PasswordHash | FullName | Bio | ProfilePic | JoinDate | NumberOfFollowe |
| 1 | johnDoe | john.doe@email.com | hashed_password_1 | John Doe | Just a regular Joe. | profilepic1.jpg | 2023-01-01 | 1 |
| 2 | janeDoe | jane.doe@email.com | hashed_password_2 | Jane Doe | Love hiking and outdoor adventures. | profilepic2.jpg | 2023-01-02 | 1 |
| 3 | coolUser123 | cooluser123@email.com | hashed_password_3 | Cool User | Always up for a cool adventure. | profilepic3.jpg | 2023-01-03 | 1 |
| 4 | travelBug | travel.bug@email.com | hashed_password_4 | Travel Bug | Exploring the world, one country at a time. | profilepic4.jpg | 2023-01-04 | 1 |
| 5 | bookworm | bookworm@email.com | hashed_password_5 | Book Worm | A room without books is like a body without a soul. | profilepic5.jpg | 2023-01-05 | 1 |

Query 10: Identify users who follow back all their followers.

Question: How can we find users who follow back every user that follows them?

```
SELECT DISTINCT f1.FollowedUserId
FROM Followers f1
INNER JOIN Followers f2 ON f1.FollowedUserId = f2.FollowingUserId AND f1.FollowingUserId = f2.FollowedUserId;
```

| Output | |
|----------------|--|
| FollowedUserId | |
| 2 | |
| 1 | |

VI.

Project demonstration

Utilizing Online SQL Compilers

The development of Connect Sphere was significantly propelled using an online SQL compiler, serving as the cornerstone for database experimentation and setup. Initially, the process involved clearing the workspace by dropping pre-existing tables, setting the stage for a fresh start. Subsequently, ten critical tables were meticulously created, laying down the structural framework of our social media platform. This foundational step was followed by populating these tables with data, simulating real-world social media interactions. Running ten specific queries then allowed us to validate the database's functionality and efficiency, ensuring the system was primed for dynamic user engagement.

Crafting with ERD Plus

ERD Plus played a pivotal role in transitioning from theoretical database design to a practical, visual schema. Having previously conceptualized an Entity-Relationship Diagram on paper, ERD Plus enabled its digital manifestation, offering insights into the art of online ER diagramming. The tool facilitated a deep dive into defining entities, attributes, and the intricate relationships between them, sharpening my understanding of database architecture. This process not only reaffirmed the database's structural integrity but also enhanced my ability to craft a coherent relational schema, essential for the seamless functionality of Connect Sphere.

Leveraging Word, Canva, and GitHub for Project Documentation and Presentation

In the final stages of the Connect Sphere project, the combined use of Microsoft Word, Canva, and GitHub proved instrumental in shaping the project's documentation, presentation, and dissemination. Microsoft Word facilitated the meticulous drafting of the project report, ensuring a detailed and structured analysis. Canva transformed this detailed content into visually engaging presentations, making complex information more accessible and understandable. GitHub then played a pivotal role in sharing the project with the world, hosting both the code and documentation for collaboration and feedback. Together, these tools underscored the importance of a holistic approach to project development, blending detailed documentation with creative presentation and open collaboration to maximize the project's impact and reach.

VII.

Self -Learning

- Embarking on the Connect Sphere project significantly deepened my understanding of database design, particularly in normalization and relational schema configuration, beyond the foundational knowledge acquired in the classroom. This endeavor required navigating advanced concepts ensure data integrity and efficiency for a social media platform's dynamic need. Without developing a GUI, my focus shifted towards backend logic, emphasizing the importance of structured information to support complex social media interactions such as user connections, content sharing, and activity tracking. This journey not only sharpened my technical skills but also highlighted the continuous

VIII.

Learning from the Project

- Working on this project allowed me to delve deeply into the foundational aspects of database management systems (DBMS), specifically tailored for a social media context. My journey was enriched by a hands-on exploration of SQL, starting with the creation of tables and mastering the nuances of inserting data into these structures. The process of populating tables with relevant, simulated social media data, from user profiles to posts and comments, gave me practical insights into how databases are structured and manipulated.
- I gained valuable experience in formulating basic to intermediate SQL queries, which are essential for retrieving, updating, and managing data efficiently. This skill set not only solidified my understanding of SQL syntax but also taught me about the importance of query optimization and database integrity.
- The project also provided a platform to apply theoretical knowledge to a practical scenario by requiring the construction of an Entity-Relationship (ER) diagram. This exercise deepened my comprehension of how entities within a database relate to one another, reinforcing concepts like cardinality and relationship types. Creating an ER diagram for a social media database showcased the interconnected nature of social platforms, from user interactions to content sharing.
- Utilizing online SQL compilers and environments was a significant aspect of my learning experience. These tools offered a flexible and accessible means to test queries, debug issues, and see immediate results of database operations. The convenience of online compilers accelerated my learning curve, providing a sandbox for experimentation and exploration without the need for extensive local setup.

IX.

Challenges Faced

- Not as such very easy project.

X.

Conclusion

The journey through the development of the ConnectSphere social media platform has been a profoundly enriching learning experience, endowing me with invaluable skills and insights into the complexities of crafting efficient database systems tailored for the dynamic world of social interactions. This project served as a practical foray into the intricate melding of database design, backend development, and the nuanced challenges of managing a digital social ecosystem.

Embracing Python for its robust backend capabilities, Flask for crafting a lightweight yet powerful web interface, and MySQL for its reliable data management, proved to be a formidable combination. These technologies together enabled the construction of a scalable and responsive platform capable of handling the multifaceted demands of social media networking. Mastering the art of establishing seamless connections between the Python environment and the MySQL database, executing sophisticated SQL queries, and dynamically rendering data within web interfaces using Flask, were among the project's core achievements.

The comprehensive understanding gained from this endeavor extends beyond mere technical prowess; it encapsulates the essential principles of database management systems, the architectural considerations vital for supporting real-time social media interactions, and the integration of various software tools to create a cohesive and user-friendly experience.

Equipped with a solid foundation in both the theoretical and practical aspects of software development, I am now more adept at navigating the complexities of building data-driven applications. This project has not only prepared me for future challenges in the technology domain but has also ignited a passion for developing solutions that facilitate meaningful digital connections.

Concluding this project, I reflect on a journey that was as much about discovery and learning as it was about development and implementation. The Connect Sphere project has been a milestone in my professional development, providing a deep dive into the world of social media platforms from a creator's perspective and offering a hands-on approach to solving the real-world challenges they present.