

|  |
| --- |
| Sarzhan Merey  **Social Media Connectme DB** |
|  |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background connectme is a modern social media platform designed to help people share content and interact with friends. Unlike other networks, it focuses on delivering highly personalized content by analyzing user activity, relationships, and preferences. For a better experience, a structured database is essential for managing interactions, friendships, and recommendations efficiently.

## Problems. Current Situation Without an optimized database, Connectme would struggle to deliver relevant content, leading to lower user engagement. poor organization may cause security vulnerabilities. Managing relationships, likes, comments, and hashtags effectively is crucial for keeping users connected and engaged.

## the Benefits of implementing a database. Project Vision A well-structured database enhances user experience by enabling faster content recommendations and secure data storage. It ensures seamless interactions and supports real-time updates. The goal is to create a dynamic and engaging social media environment where users can easily connect and share.

# Model description

## Definitions & Acronyms

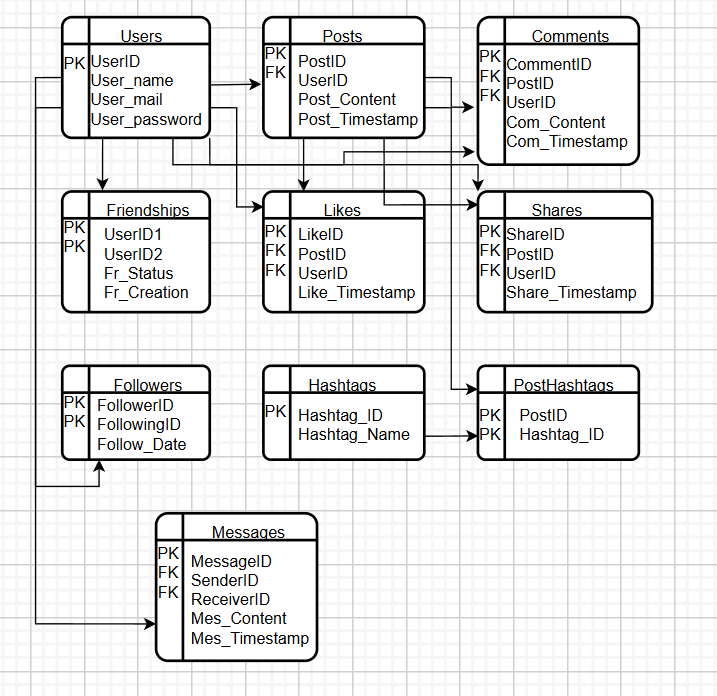
User: A registered person on the platform.

Post: Content shared by a user, including text, images, or videos.

Comment: A response to a post that adds to discussions.

Like/Share: User interactions that indicate engagement with posts.

Friendship/Following: Ways users connect and interact with each other.   
Hashtag: A keyword that helps categorize and discover posts.  
Messages: User interaction.

Logical Scheme  
  


## Logical Scheme

## Objects Connectme’s database is designed to store and manage user-generated content and interactions. The **Users** table holds personal details, while **Posts**, **Comments**, **Likes**, and **Shares** track engagement. **Friendships** and **Followers** define social connections, and **Hashtags** help categorize content. The **Messages** table facilitates private communication. This structure ensures efficient data handling, making a fast connection.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Users | UserID | Unique user identifier PK | INT |
| Users | User\_name | Username (unique) | VARCHAR |
| Users | Email | User’s email (unique) | VARCHAR |
| Users | Password | User’s password (unique) | VARCHAR |
| Posts | PostID | Unique post identifier PK | INT |
| Posts | UserID | Foreign Key (Users.UserID) | INT |
| Posts | Post\_Content | The content of the post | TEXT |
| Posts | Post\_Timestamp | Date & time of post creation | DATETIME |
| Comments | CommentID | Unique comment identifier PK | INT |
| Comments | PostID | FK of Posts(PostID) | INT |
| Comments | UserID | FK of Users(UserID) | INT |
| Comments | Com\_Content | Comment content | TEXT |
| Comments | Com\_Timestamp | Date & time of comment creation | DATETIME |
| Likes | LikeID | Unique like identifier PK | INT |
| Likes | PostID | FK, Liked post | INT |
| Likes | UserID | FK, User who liked the post | INT |
| Likes | Like\_Timestamp | Date & time of likes | DATETIME |
| Shares | ShareID | Unique share identifier PK | INT |
| Shares | PostID | FK, Shared post | INT |
| Shares | UserID | FK, User who shared the post | INT |
| Shares | Share\_Timestamp | Date & time of shares | DATETIME |
| Friendships | UserID1 | FK, first user ID | INT |
| Friendships | UserID2 | FK, second user (friend) | INT |
| Friendships | Fr\_Status | Current state of friendship request | TEXT |
| Friendships | Fr\_Creation | Date & time of created friendship | DATETIME |
| Messages | MessageID | Unique message identifier PK | INT |
| Messages | SenderID | Foreign Key of Users | INT |
| Messages | ReceiverID | Foreign Key of Users | INT |
| Messages | Mes\_Content | Text not null | TEXT |
| Messages | Mes\_Timestamp | Date & time of messages | DATETIME |
| Hashtags | Hashtag\_ID | Unique hashtag identifier PK | INT |
| Hashtags | Hashtag\_Text | Unique text | VARCHAR |
| Post\_Hashtags | PostID | FK of Posts(PostID) Part of PK | INT |
| Post\_Hashtags | Hashtag\_ID | FK of Hashtags(Hashtag\_ID), Part of PK | INT |
| Followers | FollowerID | User who follows another user. FK to Users.UserID | INT |
| Followers | FollowingID | User being followed. FK to Users.UserID | INT |
| Followers | Follow\_Date | When the following happened | DATETIME |

Comments on table relationships

**Users & Posts** → One-to-Many  
**Users & Comments** → One-to-Many  
**Users & Likes** → Many-to-Many   
**Users & Shares** → Many-to-Many  
**Users & Friendships** → Many-to-Many  
**Users & Followers** → One-to-Many  
**Users & Messages** → One-to-Many  
**Posts & Hashtags** → Many-to-Many

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UserID | User\_Name | PostID | Post\_Content | LikeID | CommentID |
| 1 | Mary | 120 | New Post | 145 | 65 |