

|  |
| --- |
| Business Template |
| **Logo / Image** |

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

## Problems. Current Situation

## the Benefits of implementing a database. Project Vision

# Model description

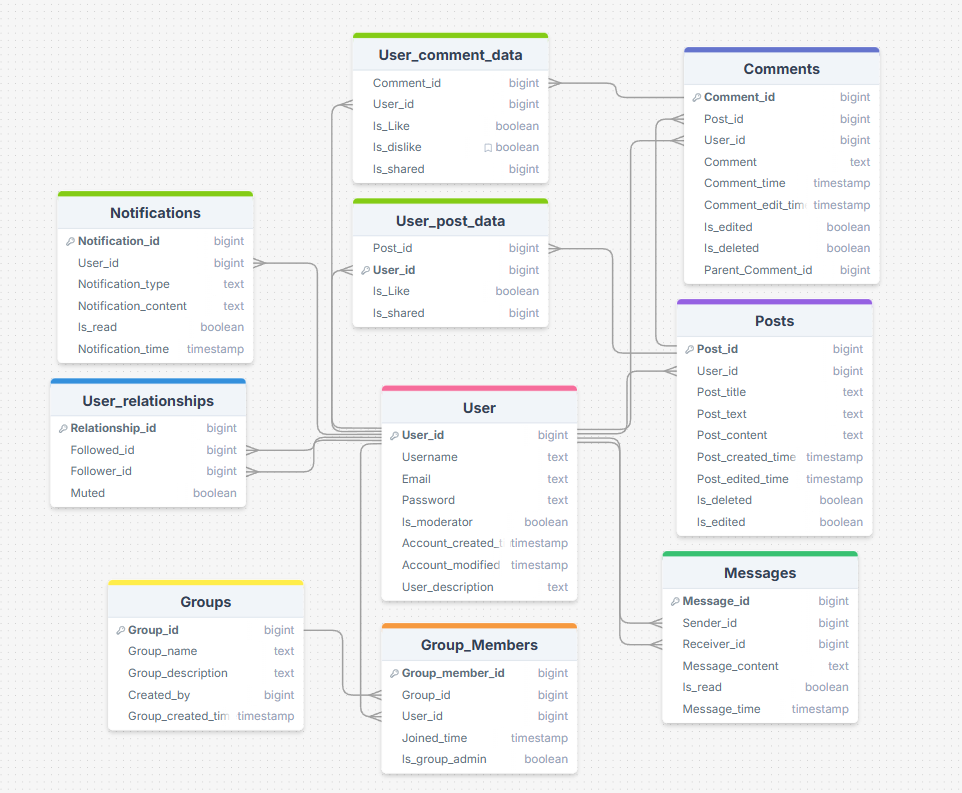
A rough sketch of how a logical scheme would look.

## 2.1 Definitions & Acronyms

The following terms and acronyms are used throughout this document:  
- PK: Primary Key  
- FK: Foreign Key  
- bigint: A data type used for storing large integer values  
- text: A data type used for storing long text strings  
- boolean: A data type used for storing binary values (True/False)

## 2.2 Logical Scheme

The logical scheme representing the database structure is shown below:



## 2.3 Objects

The following are the tables and fields included in the database:

### Table: User\_comment\_data

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| Comment\_id | bigint |  | Primary Key |
| User\_id | bigint |  | Foreign Key to User |
| Is\_Like | boolean |  | Indicates if the comment is liked |
| Is\_dislike | boolean |  | Indicates if the comment is disliked |
| Is\_shared | boolean |  | Indicates if the comment is shared |

### Table: Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Comment\_id | bigint | Not Null (automatically generated) | Primary Key |
| Post\_id | bigint |  | Foreign Key to Posts |
| User\_id | bigint |  | Foreign Key to User |
| Comment | text(100( |  | Text of the comment |
| Comment\_time | timestamp | Not Null (automatically generated) | Time when the comment was made |
| Is\_edited | boolean |  | Indicates if the comment is edited |
| Is\_deleted | boolean |  | Indicates if the comment is deleted |
| Parent\_Comment\_id | bigint |  | Parent comment id |

### Table: Posts

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Post\_id | bigint | Not Null (automatically generated) | Primary Key |
| User\_id | bigint |  | Foreign Key to User |
| Post\_title | text(100) | Not Null | Title of the post |
| Post\_text | text(200) | Not Null | Main text content of the post |
| Post\_content | text(100) |  | Content or body of the post |
| Post\_created\_time | timestamp | Not Null (automatically generated) | Creation time of the post |
| Is\_deleted | boolean |  | Indicates if the post is deleted |
| Is\_edited | boolean |  | Indicates if the post is edited |

### Table: User

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| User\_id | bigint | Not Null (automatically generated) | Primary Key |
| Username | text(50) | Not Null Unique | User's username |
| Email | text(50) | Not Null Unique | User's email address |
| Password | text(100) | Not Null | User's password |
| Is\_moderator | boolean |  | Indicates if the user is a moderator |
| Account\_created\_time | timestamp | Not Null (automatically generated) | Time when the account was created |

### Table: User\_relationships

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Relationship\_id | bigint |  | Primary Key |
| Followed\_id | bigint |  | User being followed |
| Follower\_id | bigint |  | User following another user |
| Follow | boolean |  | Indicates if the relationship is a follow |
| Muted | boolean |  | Indicates if the relationship is muted |

### Table: User\_post\_data

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Post\_id | bigint |  | Foreign Key to Posts |
| User\_id | bigint |  | Foreign Key to User |
| Is\_Like | boolean |  | Optional Like = True Dislike = False |
| Is\_shared | boolean |  | Indicates if the post is shared |

### Table: Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Group\_id | bigint | Not Null (automatically generated) | Primary Key |
| Group\_name | text(100) | Not Null | Name of the group |
| Group\_description | text(200) |  | Description of the group |
| Created\_by | bigint | Not Null | User ID of the group creator |
| Group\_created\_time | timestamp | Not Null (automatically generated) | Timestamp for when the group was created |

### Table: Group\_member\_id

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constrainst | Description |
| Group\_member\_id | bigint | Not Null (automatically generated) | Primary Key |
| Group\_id | bigint |  | Foreign Key to Groups |
| User\_id | bigint |  | Foreign Key to User |
| Joined\_time | timestamp | Not Null (automatically generated) | Timestamp for when the user joins group |
| Is\_group\_admin | boolean |  | Indicates if the user is group admin |

Relationships Descriptions:  
---------------------------

* **User to Comments**: One-to-many relationship. A user can make many comments, but each comment is made by one user.
* **User to Posts**: One-to-many relationship. A user can create multiple posts, but each post is created by one user.
* **Posts to Comments**: One-to-many relationship. A post can have many comments, but each comment belongs to one post.
* **User to User\_relationships**: Many-to-many relationship. A user can follow many other users, and each user can have many followers. The User\_relationships table tracks the follow relationships.
* **User to User\_comment\_data**: Many-to-many relationship. A user can interact (like, dislike, share) with many comments, and each comment can have multiple user interactions.
* **User to User\_post\_data**: Many-to-many relationship. A user can interact (like, dislike, share) with many posts, and each post can have multiple user interactions.
* **Groups to Group\_Members**: One-to-many relationship. A group can have many members, but each member belongs to one group.
* **User to Groups**: One-to-many relationship. A user can create many groups, but each group is created by one user.
* **Groups to User**: Many-to-many relationship (via Group\_Members). A user can be a member of many groups, and each group can have many members.
* **User to Messages**: One-to-many relationship. A user can send many messages, but each message is sent by one user (Sender\_id).
* **User to Messages (Receiver)**: One-to-many relationship. A user can receive many messages, but each message is received by one user (Receiver\_id).