The purpose of designing this database is to store and manage the data related to the networked distributed group-based client-server communication system. The database handles user information, group chat details, message data, and message states. It allows for efficient querying and updating of information, ensuring that the system remains consistent and scalable. The database consists of five tables: Users, GroupChat, UserInGroups, Messages, and MessageState.

Users Table

The Users table stores basic information about users, such as their unique identifier, name, IP address, and port. The fields in the Users table are:

* User\_ID: The primary key uniquely identifying each user.
* Name: The user's name.
* IP: The user's IP address.
* Port: The user's port number.

GroupChat Table

The GroupChat table contains information about the various group chats created within the system. The fields in the GroupChat table are:

* GroupChat\_ID: The primary key uniquely identifying each group chat.
* Coordinator: The User\_ID of the group's coordinator.

UserInGroups Table

The UserInGroups table maintains the relationship between Users and GroupChats, representing the users that are part of each group chat. The fields in the UserInGroups table are:

* UserInGroups\_ID: The primary key uniquely identifying each user-group relationship.
* User\_ID: A foreign key referencing the Users table, indicating the user in the group chat.
* Groupchat\_ID: A foreign key referencing the GroupChat table, indicating the group chat the user belongs to.

Messages Table

The Messages table stores the messages sent within group chats, along with the sender's information and timestamps. The fields in the Messages table are:

* Message\_ID: The primary key uniquely identifying each message.
* User\_ID: A foreign key referencing the Users table, indicating the sender of the message.
* Groupchat\_ID: A foreign key referencing the GroupChat table, indicating the group chat where the message was sent.
* Time: The timestamp when the message was sent.
* Message: The content of the message.

MessageState Table

* The MessageState table maintains the state of messages, indicating whether they have been acknowledged by each user in the group chat. The fields in the MessageState table are:
* MessageState\_ID: The primary key uniquely identifying each message state entry.
* User\_ID: A foreign key referencing the Users table, indicating the user whose message state is being tracked.
* Message\_ID: A foreign key referencing the Messages table, indicating the message being tracked.
* Groupchat\_ID: A foreign key referencing the GroupChat table, indicating the group chat where the message was sent.
* Message\_State: A boolean value representing whether the message has been acknowledged by the user (True or False).

The relationships between each table are as follows:

1. Users Table: This table stores information about each user in the system. It has no direct relationships with other tables but is referenced by other tables using foreign keys.
2. GroupChat Table: This table stores information about each group chat created within the system. It has a one-to-many relationship with the UserInGroups table through the GroupChat\_ID field and a one-to-one relationship with the Users table through the Coordinator field.
3. UserInGroups Table: This table represents the relationship between Users and GroupChats. It has a many-to-one relationship with both the Users table and the GroupChat table. The User\_ID field is a foreign key referencing the Users table, while the Groupchat\_ID field is a foreign key referencing the GroupChat table. Each row in this table represents a user's membership in a group chat.
4. Messages Table: This table stores messages sent within group chats. It has a many-to-one relationship with the Users table through the User\_ID field and a many-to-one relationship with the GroupChat table through the Groupchat\_ID field. Each row in this table represents a message sent by a user in a group chat.
5. MessageState Table: This table maintains the state of messages for each user in a group chat. It has a many-to-one relationship with the Users table through the User\_ID field, a many-to-one relationship with the Messages table through the Message\_ID field, and a many-to-one relationship with the GroupChat table through the Groupchat\_ID field. Each row in this table represents the state of a message for a user in a group chat.

The pattern design of the database is based on the principles of normalization and the Entity-Relationship (ER) model. The database is designed to minimize redundancy and ensure data consistency by breaking down the data into separate tables with clearly defined relationships. The ER model provides a clear and structured representation of the database schema, making it easier to understand the relationships between tables and the flow of data within the system.