**Chat App Database Outline**

Chat App is a text and voice chat application inspired by Discord. It enables real-time conversations in channels, supported by a meticulously designed database ensuring data integrity. Key entities include "Users" for authentication and relationships with channels and messages, "Roles" for user privileges, "Messages" with emojis through "Reactions," and file sharing via "Attachments" and "FileTypes." Additionally, time-limited invite links are offered through the "Invites" entity to foster user engagement.

* **Users**: records the details of each user
  + Attributes:
    - userID: int, auto\_increment, not NULL, PK
    - username: varchar(100), unique, not NULL
    - email: varchar(255), unique, not NULL
    - password: varchar(60), not NULL
    - userImageUrl: VARCHAR(255)
  + Relationships:
    - M:M relationship between Users and Servers implemented with an intersection table, UserServers, containing FKs userID and serverID
    - M:M relationship between Users and Channels implemented with an intersection table, UserChannels, containing FKs userID and channelID
    - 1:M relationship between Users and Messages implemented with userID as a FK inside Messages
* **Servers**: each Server has multiple users, and channels in which users can view or participate in real time text or voice chat
  + Attributes:
    - serverID: int, auto\_increment, not NULL, PK
    - serverName: varchar(100), not NULL
    - serverDescription: varchar(255)
    - serverImageUrl: VARCHAR(255)
  + Relationships:
    - M:M relationship between Servers and Users implemented with an intersection table, UserServers, containing FKs userID and serverID
    - 1:M relationship between Servers and Channels facilitated with serverID as a FK inside Channels
    - 1:M relationship between Servers and Invites with serverID as a FK inside Invites
* **Channels:** each Channel is part of a Server, has privileges for certain roles, or users, and a type for voice or chat
  + Attributes:
    - channelID: int, auto\_increment, not NULL, PK
    - serverID: int, not NULL, FK, ON DELETE CASCADE
    - roleID: int, not NULL, FK
    - subchannelTypeID: int, not NULL, FK
    - subchannelName: varchar(100), not NULL
  + Relationships
    - M:1 relationship between Channels and Servers facilitated with serverID as a FK inside Channels
    - M:1 relationship between Channels and Roles facilitated with roleID as a FK inside Channels
    - 1:M relationship between Channels and ChannelTypes facilitated with channelTypeID as a FK inside Channels
    - 1:M relationship between Channels and Messages facilitated with channelID as a FK inside Messages
    - M:M relationship between Channels and Users implemented with an intersection table, UserChannels, containing FKs userID and channelID
* **ChannelType**: the type of channel (voice, video)
  + Attributes:
    - channelTypeID: int, auto\_increment, not NULL, PK
    - channelType: varchar(100), unique, not NULL
  + Relationships:
    - M:1 relationship between ChannelTypes and Channels facilitated with channelTypeID as a FK inside Channels
* **Roles**: records the role for each user in a channel (creator, admin, user)
  + Attributes:
    - roleID: int, auto\_increment, not NULL, PK
    - roleName: varChar(100), unique, not NULL
  + Relationships:
    - 1:M relationship between Roles and UserServers implemented with roleID as a FK inside UserServers
    - 1:M relationship between Roles and Channels facilitated with roleID as a FK inside Channels
* **UserChannels**: the individual users that have access to a restricted Channel
  + Attributes:
    - userChannelID: int, auto\_increment, not NULL, PK, ON DELETE CASCADE
    - userID, int, not NULL, FK, ON DELETE CASCADE
    - channelID: int, not NULL, FK
  + Relationships:
    - Facilitates the M:M relationship between Users and Channels
    - M:1 relationship between UserChannels and Channels facilitated with channelID as a FK inside Userchannels
    - M:1 relationship between UserChannels and Users facilitated with userID as a FK inside UserChannels
* **UserServers**: the individual users that are members of a Channel
  + Attributes:
    - userServerID: int, auto\_increment, not NULL, PK
    - serverID: int, not NULL, FK, ON DELETE CASCADE
    - userID: int, not NULL, FK, ON DELETE CASCADE
    - roleID: int, not NULL, FK
  + Relationships:
    - Facilitates the M:M relationship between Users and Channels
    - M:1 relationship between UserServers and Users facilitated with userID as a FK inside UserServers
    - M:1 relationship between UserServers and Servers facilitated with serverID as a FK inside UserServers
    - M:1 relationship between UserServers and Roles facilitated with roleID as a FK inside UserServers
* **Attachments**: records the details of an attachment in a message
  + Attributes:
    - attachmentID: int, auto\_increment, not NULL, PK
    - messageID: int, not NULL, FK, ON DELETE CASCADE
    - fileTypeID: int, not NULL, FK
    - filename: varchar(255), not NULL
    - attachmentUrl: VARCHAR(255), not NULL
  + Relationships:
    - M:1 relationship between Attachments and Messages facilitated with attachmentID as a FK inside Messages and messageID as a FK inside Attachments
    - 1:M relationship between Attachments and fileTypes facilitated with fileTypeID as a FK inside Attachments
* **FileTypes:** records the filetype for each Attachment
  + Attributes:
    - fileTypeID: int, auto\_increment, not NULL, PK
    - fileTypeName: varchar(100), not NULL
  + Relationships:
    - M:1 relationship between FileTypes and Attachments facilitated with fileTypeID as a FK inside Attachments
* **Messages**: records the details of each message
  + Attributes:
    - messageID: int, auto\_increment, not NULL, PK
    - userID: int, FK, ON DELETE SET NULL
    - channelID: int, not NULL, FK, ON DELETE CASCADE
    - text: varchar(1024), not NULL
    - time: DateTime, not NULL
  + Relationships:
    - M:1 relationship between Messages and Users facilitated with userID as a FK inside messages
    - M:1 relationship between Messages and Channels facilitated with channelID as a FK inside Messages
    - M:1 relationship between Messages and Attachments facilitated with attachmentID as a FK inside Messages and messageID as a FK inside Attachments
    - 1:M relationship between Messages and Reactions facilitated with messageID as a FK inside Reactions
* **Reactions**: records the details of each reaction for a message
  + Attributes:
    - reactionID: int, auto\_increment, not NULL, PK
    - emojiID: int, auto\_increment, not NULL, FK
    - userID: int, auto\_increment, not NULL, FK, ON DELETE CASCADE
    - messageID: int, auto\_increment, not NULL, FK, ON DELETE CASCADE
  + Relationships:
    - M:1 relationship between Reactions and Messages facilitated with messageID as a FK inside Reactions
    - M:1 relationship between Reactions and Emojis facilitated with enojiID as a FK inside Reactions
* **Emojis**: records the details of each Emoji for a Reaction
  + Attributes:
    - emojiID: int, auto\_increment, not NULL, PK
    - emojiCode: varchar(255), unique, not NULL
    - emojiName: varchar(100), not NULL
  + Relationships:
    - 1:M relationship between Emojis and Reactions facilitated with emojiID as a FK inside Reactions
* **Invites:** records the invite-related data for each invite
  + Attributes:
    - inviteID: int, auto\_increment, not NULL, PK
    - serverID: int, not NULL, FK, ON DELETE CASCADE
    - inviteCode: varchar(255), unique, not NULL
    - expirationTime: DateTime, not NULL
  + Relationships:
    - M:1 relationship between Invites and Servers with serverID as a FK inside Invites