Database Schema Documentation

Overview

This document describes the database schema for the messaging app, including all tables, fields, relationships, and indexing strategies. The database is designed to ensure scalability, efficiency, and security.

Database Technology

We are using **PostgreSQL** for its robust support for relational data, JSON storage for flexible data types, and indexing features.

Tables and Relationships

1. Users Table

- **Description**: Stores user information.
- **Fields**:

Field	Type	Constraints	Description
`id`	UUID	Primary Key	Unique identifier for a user.
`name`	VARCHAR(255)	NOT NULL	Full name of the user.
`email`	VARCHAR(255)	UNIQUE, NOT NULL	User's email address.
`password`	VARCHAR(255)	NOT NULL	Hashed password.
`avatar`	TEXT	NULLABLE	URL to the user's avatar.
`created_at`	TIMESTAMP	DEFAULT NOW()	Timestamp of account creation.

- **Indexes**:
  - `UNIQUE(email)`: For fast lookups and ensuring unique emails.

---

2. Chats Table

- **Description**: Stores chat metadata.
- **Fields**:

Field	Type	Constraints	Description
`id`	UUID	Primary Key	Unique identifier for the chat.
`created_at`	TIMESTAMP	DEFAULT NOW()	Timestamp of chat creation.

---

3. Participants Table

- **Description**: Links users to chats.
- **Fields**:

Field	Type	Constraints	Description
`id`	UUID	Primary Key	Unique identifier for the participant entry.
`chat_id`	UUID	Foreign Key (Chats)	The chat this user belongs to.
`user_id`	UUID	Foreign Key (Users)	The user in the chat.

- **Relationships**:
  - `chat_id` → `Chats(id)`: One-to-many.
  - `user_id` → `Users(id)`: One-to-many.

---

4. Messages Table

- **Description**: Stores messages exchanged in chats.
- **Fields**:

Field	Type	Constraints	Description
`id`	UUID	Primary Key	Unique identifier for the message.
`chat_id`	UUID	Foreign Key (Chats)	The chat this message belongs to.
`sender_id`	UUID	Foreign Key (Users)	The user who sent the message.
`content`	TEXT	NOT NULL	The message content.
`type`	ENUM('text', 'image', 'video', 'file')	NOT NULL	Type of the message.
`timestamp`	TIMESTAMP	DEFAULT NOW()	When the message was sent.

- **Indexes**:
  - `INDEX(chat_id, timestamp)`: For efficient message retrieval in chats.

---

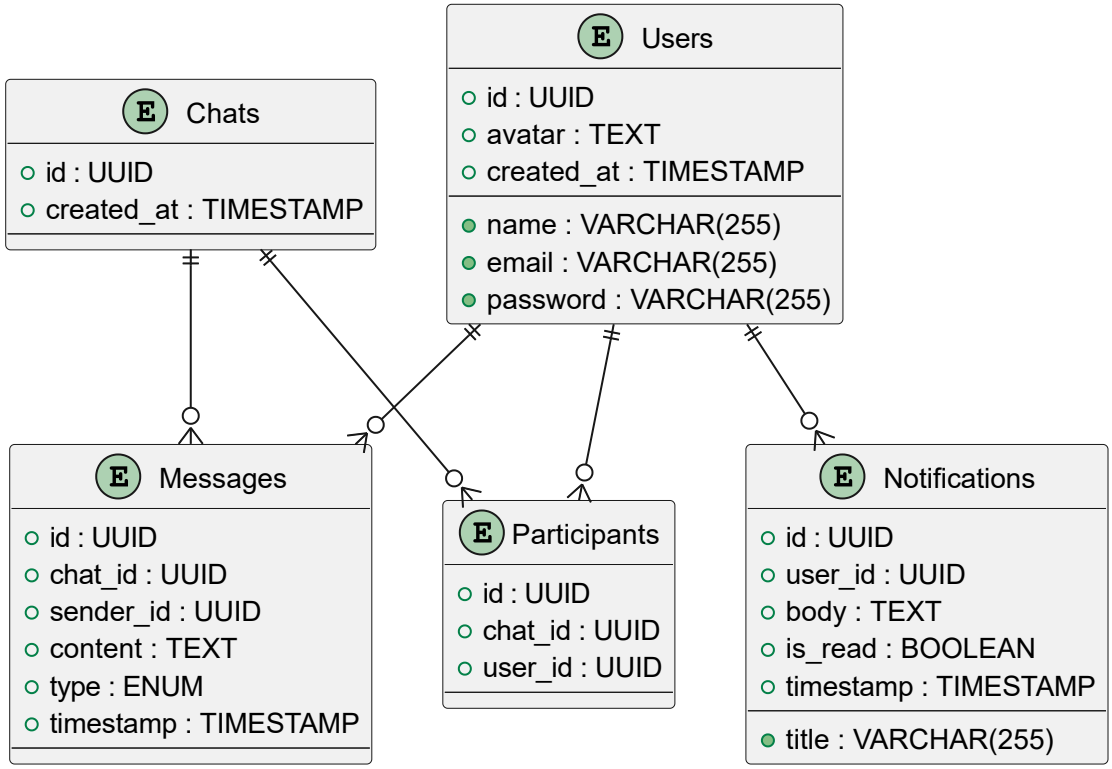
5. Notifications Table

- **Description**: Stores notification data for users.
- **Fields**:

Field	Type	Constraints	Description

Field	Type	Constraints	Description
id	UUID	Primary Key	Unique identifier for the notification.
user_id	UUID	Foreign Key (Users)	The user to whom the notification is sent.
title	VARCHAR(255)	NOT NULL	Title of the notification.
body	TEXT	NOT NULL	Notification content.
is_read	BOOLEAN	DEFAULT FALSE	Whether the notification is read.
timestamp	TIMESTAMP	DEFAULT NOW()	When the notification was created.

Entity Relationship Diagram (ERD)



Notes for Database Schema

1. Use UUID for all primary keys to ensure globally unique identifiers.
2. Encrypt user passwords using a strong hashing algorithm like bcrypt.
3. Normalize data to reduce redundancy but use JSON fields for flexible data types if required (e.g., for message attachments).
4. Set up database replication for scalability and fault tolerance.
5. Create backups regularly and monitor performance using indexing and query optimization.