Database Design

Lizzie Maczka

November 3, 2023

Overview

For my capstone project, I have chosen to employ a relational database model, specifically PostgreSQL. I have chosen this database model based on the nature and requirements of my web application Character Notebook. The data involved in Character Notebook will be structured with clear relationships between entities, so a relational database model is suitable for the application.

Database Structure

Users table:

- User_id: primary key, serial
- Username: string
- Password: string
- Email: string

Characters table:

- Character_id: primary key, serial
- Name: string

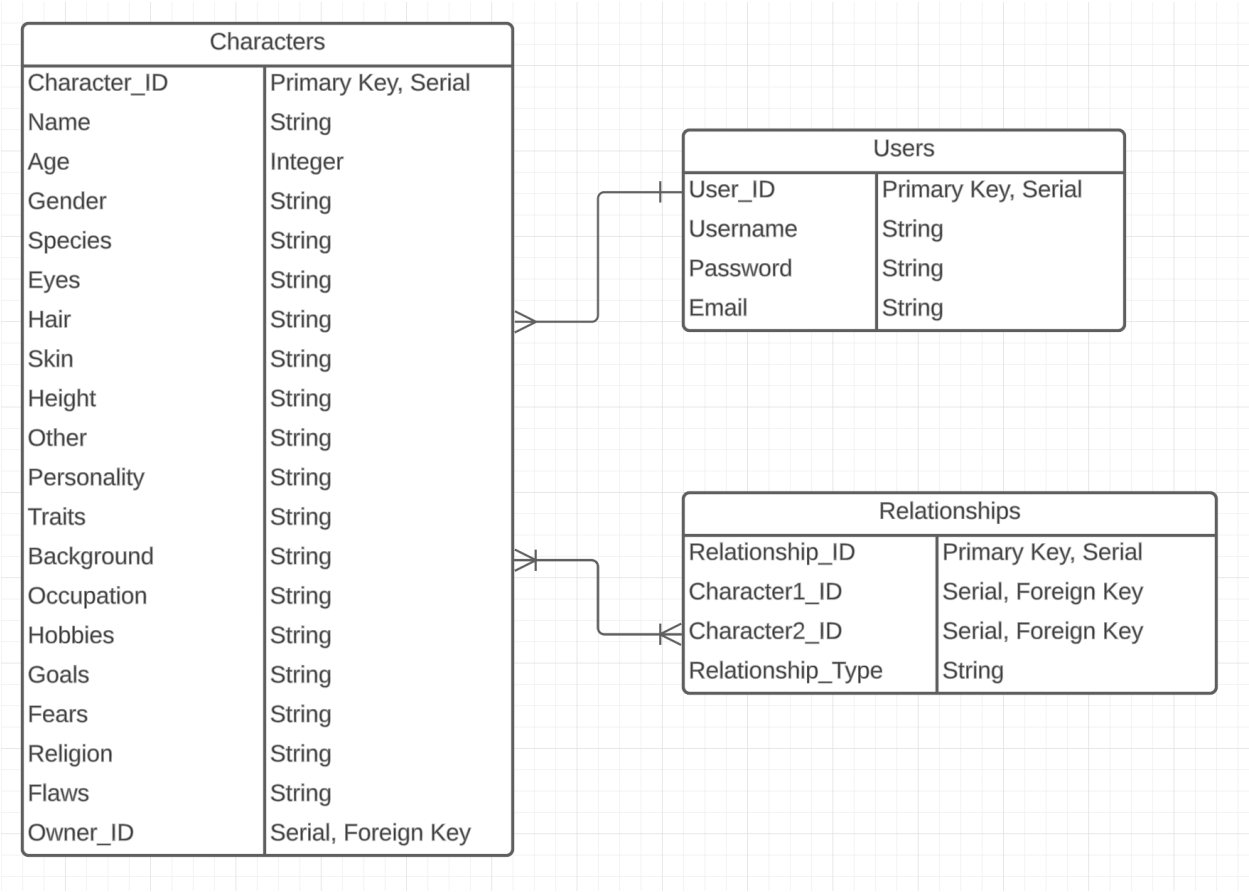
- Age: integer
- Gender: string
- Species: string
- Eyes: string
- Hair: string
- Skin: string
- Height: string
- Other: string
- Personality: string
- Traits: string
- Background: string
- Occupation: string
- Hobbies: string
- Goals: string
- Fears: string
- Religion: string
- Flaws: string
- Owner_ID: foreign key, serial, references the Users table

Relationships table

- Relationship_id: Primary key, serial
- Relationship_type: string
- Character1_ID: foreign key, serial, references the characters table

- Character2_ID: foreign key, serial, references the characters table

Entity-Relationship Diagram



In this diagram, the Users table and Characters table have a one-to-many relationship. One user can own many characters. The Characters and Relationships tables share a many-to-many relationship. Many characters can have many relationships, and relationships are shared between two characters.

Implementation and Interactions

Users:

Users will create accounts to use Character Notebook. The registration process will require a username, password, and email address. The login process will only require the username and password. This information will be stored in the Users table. When users login, the username and password they enter will be compared to the username and password associated with their account in the database. After users successfully login, they will be able to view and create characters or view relationships. When users create their characters, the character profiles data will be stored within the Characters table, however, each character will be associated with the user through the Owner_ID.

Characters:

When users login, they will be presented with the option to view their characters or the relationships between them. The View Characters option will consist of a list of the characters the user has created as well as the options to search through the list for a specific character or to create a new character. When creating a new character or editing an existing one, the user will be presented with a template consisting of fields to fill out in regards to the character's attributes. The information entered into these fields will be stored within the Characters table.

Relationships:

After users have created at least two characters, they will be able to set the relationship between the characters. There will be four options for the user to choose from: friends, family, lovers, and enemies. The relationship type as well as the two characters involved will be stored in the Relationships table. The characters will be associated with the Characters table through the Character1_ID and Character2_ID. After discerning the relationships between characters, users will be able to view a visualized map of all the relationships they have created. The data for this map will be pulled from the Relationships table and compiled into a visual format.