**2.1 Design: Database**

1. **Database Technology Choice:**

For CineLit, a relational database like PostgreSQL is the most suitable choice. This decision is based on the following factors:

* **Highly normalized data:** The application will handle data with multiple relationships, such as users, books, movies, reviews, and groups.
* **Many-to-many relationships:** For instance, users can be part of multiple groups, and groups can have multiple users. Similarly, users can follow multiple other users and be followed by many users.
* **Cascading updates:** As users update their profiles or create new reviews, these changes need to be reflected consistently across the application.

1. **Data Structures (Tables and Columns):**

* **Users:** id (Primary Key, Serial), username (Varchar), email (Varchar), password (Varchar), profile\_picture (Varchar), bio (Text)
* **Books:** id (Primary Key, Serial), title (Varchar), author (Varchar), genre (Varchar), cover\_image (Varchar), publication\_year (Integer)
* **Movies:** id (Primary Key, Serial), title (Varchar), director (Varchar), genre (Varchar), poster\_image (Varchar), release\_year (Integer)
* **Reviews**: id (Primary Key, Serial), user\_id (Foreign Key, Integer), content\_type (Varchar), content\_id (Integer), rating (Integer), review\_text (Text), spoiler\_warning (Boolean), timestamp (Timestamp)
* **Groups:** id (Primary Key, Serial), name (Varchar), description (Text), created\_by (Foreign Key, Integer), creation\_date (Date)
* **Group\_Members:** group\_id (Foreign Key, Integer), user\_id (Foreign Key, Integer)
* **Followers**: follower\_id (Foreign Key, Integer), following\_id (Foreign Key, Integer)

1. **Usage of Data Structures in the Application:**

* **Users:** This table stores user information, such as username, email, password, profile picture, and bio. It is crucial for user authentication, profile management, and displaying user-related content like reviews and group memberships.
* **Books:** This table stores information about books, including title, author, genre, cover image, and publication year. It is used to display book details, search results, and recommendations.
* **Movies**: This table stores information about movies, including title, director, genre, poster image, and release year. It is used to display movie details, search results, and recommendations.
* **Reviews:** This table stores user reviews for both books and movies, including user\_id, content\_type (book or movie), content\_id (book or movie id), rating, review text, spoiler warning, and timestamp. It is used to display reviews, aggregate ratings, and generate personalized recommendations.
* **Groups:** This table stores information about user-created groups, including name, description, creator, and creation date. It is used for displaying group details, managing memberships, and facilitating discussions among group members.
* **Group\_Members:** This table represents the many-to-many relationship between users and groups. It is used to manage group memberships and display the groups a user belongs to.
* **Followers:** This table represents the many-to-many relationship between users who follow and are followed by others. It is used to manage user connections and display followers and following lists on user profiles.

By designing the database with these tables and relationships, CineLit can efficiently store and manage the data required for its core functionalities, such as user authentication, content discovery, reviews, and social interactions. This design also ensures optimal performance and scalability for the application's future growth.