|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Goodreads Database Theory and Design  Group 11  Charlie Vandel, Harsha Gudem, Madalyn Young & Sakshi Pandey  IS 6420-001 Database Theory and Design  Stanford Pugsley  10/24/2023 |
|  |

Table of Contents

[Executive Summary 2](#_Toc149001332)

[General Description 2](#_Toc149001333)

[Prioritized Requirements 3](#_Toc149001334)

[New Venture 4](#_Toc149001335)

[Conceptual Model 4](#_Toc149001336)

[Logical Model 5](#_Toc149001337)

[Physical Model 6](#_Toc149001338)

[Requirement Review 6](#_Toc149001339)

[Ethical Considerations 7](#_Toc149001340)

[Conclusion and Next Steps 8](#_Toc149001341)

[References 8](#_Toc149001342)

[SQL Statements - Create Tables: 9](#_Toc149001343)

[SQL Tables 13](#_Toc149001344)

[Figure 1: Add\_a\_book table: 13](#_Toc149001345)

[Figure 2: Book\_author table: 13](#_Toc149001346)

[Figure 3: User\_group\_table\_join table: 14](#_Toc149001347)

[Figure 4: Book\_review table: 14](#_Toc149001348)

[Figure 5: Interaction\_type table: 14](#_Toc149001349)

[Figure 6: Interaction table: 15](#_Toc149001350)

[Figure 7: User\_group table: 15](#_Toc149001351)

[Figure 8: Author table: 15](#_Toc149001352)

# Executive Summary

Reading has always been an important part of society. With advances in technology, reading has become something you can do on the way to work via an audiobook or waiting in line at a store through an app on your phone. Goodreads creates a space where reading does not have to be an individualized activity, but a community event.

Goodreads emphasizes creating a better experience for book lovers to discover and discuss books. It has built up a collection of thousands of books that have been published throughout time for users to find and interact with. The company has created a universal method for evaluating books and sharing them with other book lovers. Goodreads was the future of book reviews, book sharing, and reading in general when it was founded in 2006. Now in 2023, there are new websites vying for the same user platform that Goodreads has maintained. Our group wanted to evaluate this company as we saw a well-established company with clear relations in its activities but also has significant room for growth.

To build out the Goodreads database, we found the focus needed to be both on the User and the Book. Almost every other function of the website can tie back to these two entities. To create the appropriate features, we created sixteen tables: Goodreads\_User, Book, AddABook, BookAuthor, Author, Book\_Edition, BookGenre, Genre, Book\_Review, UserCreatedShelves, Default\_Shelf\_Code, DefaultShelf, UserGroupTableJoin, User\_Group, Interaction, and InteractionType.

Once the tables were created, we created dummy data to fill in the tables. Though the data is not real, it reflects what we would expect to see in this database in real life. Though it is a small sample of data, it clearly exemplifies how each of the tables interact with each other and how the website itself can function so the user can have a better reading experience on the front-end of the site.

Goodreads is an amazing website for readers of all kinds. As we built these models and interacted with the interface, we felt there were so many opportunities to make Goodreads an even better reader database. We would be excited to see these ideas be executed and created relations for what we anticipate the scope of the project would entail on the database side.

# General Description

Goodreads, a subsidiary of Amazon, is a social media website centered around a database of books. It was founded in 2006 and by the time it was acquired in 2013, it had reported over 90 million users. Goodreads mission statement is “to help readers discover books they love and get more out of reading.” However, since its acquisition, Amazon has not given much if any effort to maintain or improve Goodreads and the website has remained relatively the same for the last decade.

The basic structure of Goodreads is that individuals can sign up for a Goodreads account which allows them the ability to catalog reading lists. If a user has read a book, the user can mark that book as read, add it to a customized shelf (i.e., Favorites-Shelf) , rate it, and provide a detailed review. Other users have complete visibility of all ratings and reviews of a particular book. The social portion of Goodreads comes in through the reviews and reading progress updates. Users can interact with other user reviews through likes, comments, and shares. Users can also join reading groups, similar to an online book club. This allows users to become connected through not only the books read, but also through the interactions surrounding those books.

If Goodreads had no users, it could still exist as a catalog of books. Goodreads has an extensive database of books detailing the various editions, covers, languages, format type, and publication information. Along with this, Goodreads stores author information and biographies. Goodreads allows users to connect and follow authors to get updates on their future publications.

Database tables are a critical component in the functionality of the Goodreads website. At the core, Goodreads needs the ability to accurately catalog books and all the details a book could have, such as ISBN numbers, synopsis, number of pages, and publication house. If Goodreads did not have the functionality to browse books, there would be no draw for users. Goodreads also needs to be able to allow users to have an individualized experience with shelving and tracking books. The website can handle this by creating database tables for default - auto generated - shelves as well as a database table to house any bookshelf a user would like to create. These shelves have the ability to contain any of the books within the Goodreads book datatables so the users can track the books they have read and rated in their own separate profile. Because users interacting with the book database and other users is a key component of the website, Goodreads also has to make and maintain database tables for user information.

# Prioritized Requirements

When researching the services that Goodreads provides, it became apparent that the company is first and foremost a place for people to interact with books. So all the relations and tables that we created would be related to users and to books.

* Users can review and rate books.
* Users have books on one of three shelves, “to read”, “currently reading” and “read”.
* Users have groups.
* Books have one or more authors, genres, and multiple editions.

After evaluating the requirements, we decided that the following tables would be created:

* User\_group
* User\_group\_table\_join
* Default\_shelf\_code
* Default\_shelf
* User\_created\_shelves
* User
* Book\_genre
* Genre
* Book\_edition
* Book
* Book\_author
* Author
* Add\_a\_book
* Book\_review
* Interaction
* interaction\_type

By the end of the project, we created tables for each of the requirements listed above. With this foundation in place, we can easily add more tables for any new ventures in the future.

# New Venture

With a strong user base but few improvements or updates to the service in years, there are many potential new business areas the company can expand into. The most lucrative would be to start a subscription service which gives access to premium features, including exclusive author content, first looks at new books and reader analytics. In the entity relationship diagram, we would add the entities “subscription”, “subscription\_access”, and “reader\_analytics”. The subscription entity would have the user\_id, their subscription status, and the date the subscription began. The subscription\_access entity would include boolean values of the specific premium features included in the subscription. The reader\_analytics entity would include the user\_id, the average time spent on each book and the number of books read each year.

Companies in nearly every sector have attempted to implement a subscription model to their services, which generates steady and consistent profits on the same service. Offering an optional monthly fee targeted at customers interested in the premium features would increase profits and in turn allow Goodreads to reinvest the new income into new services that were otherwise inaccessible. This would then attract new users and stimulate the growth cycle.

# Conceptual Model

A diagram of a computer model

Description automatically generated

The first step in creating the database was to create a conceptual model, allowing better organization and understanding of the connections between required tables. As mentioned in the Prioritized Requirements, the database consists of two main sections, one relating to the user and one relating to the books. Thus, each relation in the conceptual model either connected these two sections or expanded upon them. For example, the default\_shelf relation organizes user’s books into “to read”, “currently reading” and “read” shelves. This is a disjoint relationship because a user can only have a book on one of those three shelves, there is no option to store the book anywhere else. Also, the book\_edition relation provides more information on each book.

As seen in the model above, a majority of the relationships are many to many because there are few general restrictions between users, books and their related entities. One book can have many authors, and authors can have multiple books for example. Also, users can be in multiple groups, and groups can have multiple users. One challenge this poses is when normalizing the conceptual model to the logical model, relations will need to be added to act as intermediary on the many to many relationships. Despite this, using the conceptual model to quickly jot down and organize the requirements for the Goodreads database was a useful starting point.

# Logical Model

**A diagram of a computer network

Description automatically generated**

In the progression from the conceptual model to the more intricate relational model, we have advanced to the second phase of the database development process. This relational model offers a comprehensive and detailed representation of the database that we aim to construct. A key feature of this model is the deliberate identification of primary keys and foreign keys within each table, a practice that significantly bolsters data integrity.

One of our approaches has been to address binary many-to-many relationships by establishing distinct relations wherein the primary key is a composite of foreign keys derived from the associated entities. For instance, the "User\_ID" serves as the primary key in the "User" table and as a foreign key in several others, including "Book\_Review," "Interaction," "AddABook," "UserGroupTableJoin," "DefaultShelfCode," and "UserCreatedShelf" tables. This method, applied consistently, ensures a well-structured and normalized database.

Additionally, we have conscientiously structured separate relations for multi-valued properties, such as genre. The relational model not only provides guidance on the physical establishment of table connections during the database construction but also places a strong emphasis on the reduction of data redundancy. This design enables users to seamlessly insert, delete, and modify data without incurring inconsistencies or anomalies.

Our approach prioritizes the creation of a robust and user-friendly database, meticulously adhering to the highest standards of data management. The resulting database is engineered to meet the most stringent data integrity and usability criteria.

# Physical Model

# Requirement Review

At the conclusion of the project, we successfully realized and operationalized all our prioritized features within our database. These features encompass:

* User Account Creation: Users can establish their Goodreads accounts.
* Diverse Book Selection: Books of various genres are accessible to users.
* Reader Profiles and Bookshelves:
  + Users have the capability to craft and personalize reader profiles.
  + Readers can either add books to default shelves or create custom ones.
  + Personal bookshelves enable the organization and classification of books, including categories like "Read," "Want to Read," and "Currently Reading"
* Social Interaction:
  + Readers can connect with fellow readers, track their reviews, and engage with others on the platform.
  + User-generated content sharing, including book reviews and ratings, is facilitated.
* Author Profiles Access: Readers have the privilege of accessing author profiles, featuring biographies and lists of their literary works.

To implement these features, we identified the creation of specific tables, each of which was successfully completed:

* goodread\_user: Containing 100 distinct records.
* book\_edition
* book
* Book\_genre- 10 unique genres
* user\_created\_shelves
* Default\_shelf- 3 system generated shelfs
* add\_a\_book
* author
* book\_author
* user\_group
* user\_group\_table\_join
* book\_review
* interaction\_type
* interaction

 We were able to complete every one of these tables. *(See Figure ABC in Appendix- to be updated)*

# Ethical Considerations

Goodreads is committed to maintaining a high standard of ethics and responsible data management. While the risk of ethical issues in the Goodreads database is currently low, we must outline our ethical considerations to ensure our platform remains a trusted and secure space for book enthusiasts. These considerations encompass:

* **User Privacy and Data Security:** We must prioritize our users' privacy and data security. We must always seek user consent for data collection and usage.
* **Transparency**: must be open and transparent about how user data is utilized and shared and provide users with easy access to the privacy policy and terms of service.
* **Content Moderation:** Goodreads must uphold community guidelines to ensure that user-generated content on Goodreads is respectful and follows our platform's rules. Content moderation must be carried out with fairness and consistency.
* **User Consent for Recommendations:** If we introduce features like book recommendations based on user preferences, we will always seek explicit user consent to provide these recommendations. Users will have control over the data they share for this purpose.
* **Ethical Data Handling:** We must only use data for ethical purposes that benefit our users and enhance their experience on Goodreads. Data must not be exploited for unethical or harmful purposes.

As we continue to develop and evolve the Goodreads platform, these ethical considerations will guide our actions and decisions, ensuring that the platform remains a trusted and ethical space for book lovers to discover, connect and share their passion for literature.

# Conclusion and Next Steps

Goodreads is a widely used platform for book enthusiasts, providing a space for readers to discover, review, and connect over their favorite books and authors. In this project, we delved deeper into the functionalities of Goodreads and constructed a comprehensive database that encapsulates the features it offers. Our analysis underscores the value of this application, and we anticipate its enduring popularity and continuous enhancement for its dedicated user base.

Looking ahead, our project has the potential for an expanded database, introducing additional entity types and data. One avenue for expanding our data is to encompass a broader range of book genres, broadening the horizons of our catalog. Some prospective features and requirements we may consider for future development on Goodreads are as follows:

* Recommended Books: Users could specify their preferred book genres and authors when signing up on Goodreads. The platform could then offer tailored book recommendations based on these preferences. With more user interactions, these recommendations would become increasingly precise. While this feature was not incorporated into our current database due to its complexity, we recognize its potential. The challenge lies in understanding whether the recommendations should solely stem from Goodreads' algorithms or if user-generated recommendations from friends should also be integrated. Given more time, this is an aspect we could explore further.
* Subscription Service for Premium Features: Introduce a premium subscription model for users who want enhanced features. The subscription could include exclusive author content, early access to new book releases, advanced reader analytics, ad-free browsing, and other premium benefits. This could help generate revenue and sustain the platform's growth.
* Book Radio: Goodreads could potentially introduce a "Book Radio" feature that allows users to listen to audiobooks, author interviews, and literary discussions. This feature was not included in our initial database due to the wealth of existing features we needed to address. However, in future iterations, we could consider adding "Book Radio" as a distinct entity type in our database, enhancing the reading experience by incorporating audio content.
* Add a new DNF shelf type: At present, when a user starts reading a book and decides not to finish it, they have two options: to rate the book with a star rating, which may not reflect their true opinion, or to leave it unrated. This often results in users feeling compelled to assign an arbitrary rating or neglecting books they didn't complete, thus skewing the overall rating system and affecting the accuracy of recommendations. The addition of a "Did Not Finish" default shelf, abbreviated as DNF, would be a game-changer for Goodreads. Here's how it could work:

As we continue to develop and refine the Goodreads platform, these potential enhancements could further enrich the reading experience for our users, promoting greater book discovery, community interaction, and engagement. Goodreads remains committed to providing a dynamic and user-focused environment for book lovers worldwide.

# References

* [Goodreads, Amazon's website for book lovers, causes problems in publishing - The Washington Post](https://www.washingtonpost.com/technology/2023/07/01/amazon-goodreads-elizabeth-gilbert/)
* [Mockaroo](http://www.mockaroo.com)- Dummy data

# SQL Statements - Create Tables:

**drop table if exists interaction;**

**drop table if exists interaction\_type;**

**drop table if exists book\_review;**

**drop table if exists user\_group\_table\_join;**

**drop table if exists user\_group;**

**drop table if exists book\_author;**

**drop table if exists author;**

**drop table if exists add\_a\_book;**

**drop table if exists default\_shelf\_code;**

**drop table if exists default\_shelf;**

**drop table if exists user\_created\_shelve;**

**drop table if exists book\_genre;**

**drop table if exists genre;**

**drop table if exists book;**

**drop table if exists book\_edition;**

**drop table if exists goodreads\_user;**

**create table goodreads\_user (**

**user\_id integer not null,**

**first\_name varchar(150) not null,**

**last\_name varchar(150) not null,**

**email varchar(250) not null,**

**birth\_date date,**

**join\_date date not null, --generated automatically**

**last\_active\_date date not null,**

**is\_author boolean not null,**

**primary key (user\_id)**

**);**

**create table book\_edition (**

**ISBN varchar not null,**

**book\_title varchar (1000) not null,**

**language varchar(100),**

**format varchar(100),**

**page\_number integer,**

**published\_date date,**

**PRIMARY KEY (ISBN)**

**);**

**create table book (**

**book\_id integer not null,**

**ISBN varchar not null,**

**book\_title varchar(1000) not null,**

**average\_rating NUMERIC,**

**constraint book\_fk\_ISBN foreign key(ISBN) references book\_edition(ISBN),**

**primary key(book\_id)**

**);**

**create table genre (**

**genre\_id integer not null,**

**genre\_name varchar (100) not null,**

**PRIMARY KEY (genre\_id)**

**);**

**create table user\_created\_shelve(**

**shelf\_id integer not null,**

**shelf\_name varchar(250) not null,**

**user\_id integer not null,**

**constraint user\_created\_shelve\_fk\_user\_id foreign key(user\_id) references goodreads\_user(user\_id),**

**primary key (shelf\_id)**

**);**

**create table default\_shelf (**

**default\_shelf\_id integer not null,**

**default\_shelf\_name varchar(150) not null,**

**primary key (default\_shelf\_id)**

**);**

**create table default\_shelf\_code (**

**default\_shelf\_code integer not null,**

**default\_shelf\_id integer not null,**

**user\_id integer not null,**

**primary key (default\_shelf\_code),**

**--constraint default\_shelf\_id check (default\_shelf\_id in (1,2,3)),**

**constraint default\_shelf\_code\_fkey\_default\_shelf\_id foreign key (default\_shelf\_id) references default\_shelf (default\_shelf\_id)**

**);**

**create table add\_a\_book (**

**user\_id integer,**

**book\_id integer not null,**

**default\_shelf\_id integer not null,**

**shelf\_id integer not null,**

**primary key (user\_id, book\_id, default\_shelf\_id, shelf\_id),**

**constraint addabook\_fk\_user\_id foreign key(user\_id) references goodreads\_user(user\_id),**

**constraint addabook\_fk\_book\_id foreign key(book\_id) references book(book\_id),**

**constraint addabook\_fk\_default\_shelf\_id foreign key(default\_shelf\_id) references default\_shelf(default\_shelf\_id),**

**constraint addabook\_fk\_shelf\_id foreign key(shelf\_id) references user\_created\_shelve(shelf\_id)**

**);**

**create table author(**

**author\_id integer primary key,**

**author\_name Varchar(1000),**

**biography varchar(3000)**

**);**

**create table book\_author(**

**book\_id integer primary key,**

**author\_id integer,**

**constraint book\_author\_fk\_book\_id foreign key(book\_id) references book(book\_id),**

**constraint book\_author\_fk\_author\_id foreign key(author\_id) references author(author\_id)**

**);**

**create table user\_group (**

**group\_id integer primary key,**

**group\_name varchar not null,**

**bookshelf varchar(100) not null,**

**tags varchar(100) not null**

**);**

**create table user\_group\_table\_join (**

**user\_id integer,**

**group\_id integer,**

**joindate date not null,**

**constraint usergrouptable\_fk\_user\_id foreign key(user\_id) references goodreads\_user(user\_id),**

**constraint usergrouptable\_fk\_group\_id foreign key(group\_id) references User\_group(group\_id),**

**primary key(user\_id,group\_id)**

**);**

**create table book\_review (**

**review\_id integer not null,**

**user\_id integer,**

**book\_id integer,**

**rating integer not null,**

**review varchar(100),**

**spoilers boolean,**

**date\_started date,**

**date\_completed date,**

**constraint book\_review\_fk\_user\_id foreign key(user\_id) references goodreads\_user(user\_id),**

**constraint book\_review\_fk\_book\_id foreign key(book\_id) references book(book\_id),**

**PRIMARY KEY (review\_id)**

**);**

**create table interaction\_type(**

**interaction\_type\_id integer primary key,**

**reaction\_name varchar**

**);**

**create table interaction(**

**review\_id integer primary key,**

**user\_id integer,**

**interaction\_type\_id integer,**

**constraint interactiontable\_fk\_reviwe\_id foreign key(review\_id) references book\_review(review\_id),**

**constraint interactiontable\_fk\_user\_id foreign key (user\_id) references goodreads\_user(user\_id),**

**constraint interactiontable\_fk\_interaction\_type\_id foreign key (interaction\_type\_id) references interaction\_type(interaction\_type\_id)**

**);**

**alter table book**

**ADD CONSTRAINT book\_unique\_book\_id UNIQUE (book\_id);**

**create table book\_genre (**

**book\_id integer not null,**

**genre\_id integer not null,**

**PRIMARY KEY (book\_id,**

**genre\_id),**

**CONSTRAINT book\_genre\_fkey\_book\_id FOREIGN KEY (book\_id) REFERENCES book(book\_id),**

**CONSTRAINT book\_genre\_fkey\_genre\_id FOREIGN KEY (genre\_id) REFERENCES genre(genre\_id)**

**);**

# SQL Tables- Harsha to add the remaining tables

## Figure 1: Add\_a\_book table:

**A screenshot of a computer

Description automatically generated**

## Figure 2: Book\_author table:

**A screenshot of a computer screen

Description automatically generated**

## Figure 3: User\_group\_table\_join table:

**A screenshot of a graph

Description automatically generated**

## Figure 4: Book\_review table:

**A screenshot of a computer

Description automatically generated**

## Figure 5: Interaction\_type table:

**A screenshot of a computer

Description automatically generated**

## Figure 6: Interaction table:

**A screenshot of a computer

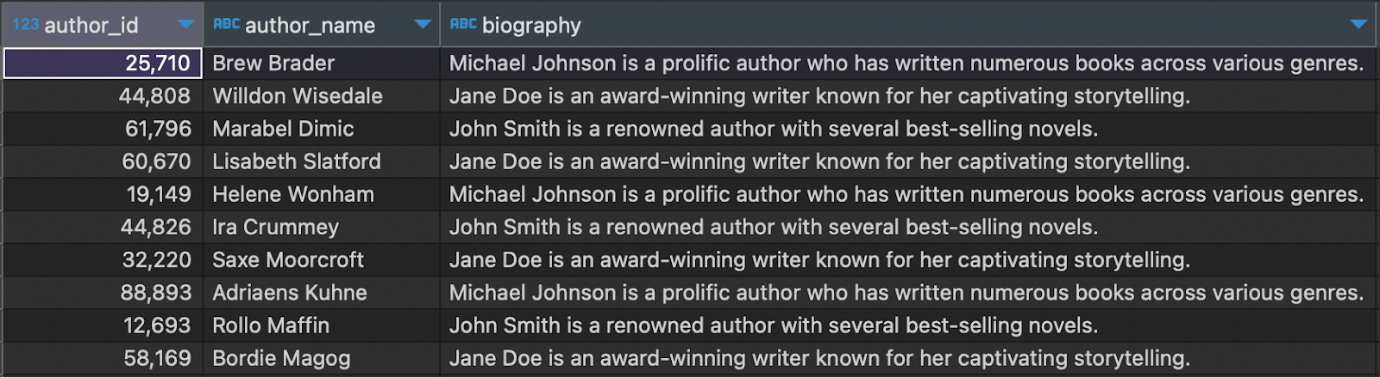
Description automatically generated**

## Figure 7: User\_group table:

**A screen shot of a computer

Description automatically generated**

## Figure 8: Author table:

****