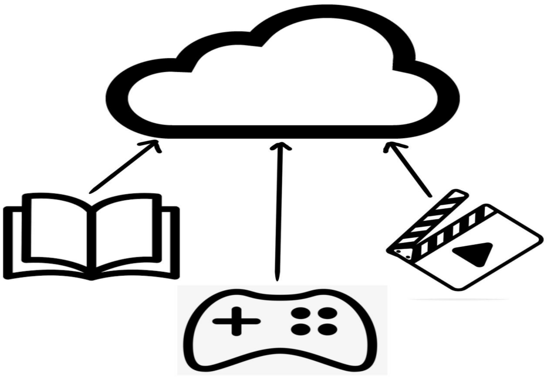


MediaTrak
MediaCorp, Inc.
Group #1: Justin Doornbos, Tom Zheng

I. Overview

As members of the group, MediaCorp, we propose a program called the MediaTrak. The objective of this program is to allow users to keep track of their media collections under three categories including books, games, and movies. This program is similar to any inventory system that keeps data log for the user and due to the increasing production of media, users who are experiencing the difficulty of remembering what titles they have played, watched, or read could effectively use this create their own personal all-in-one digital media library.

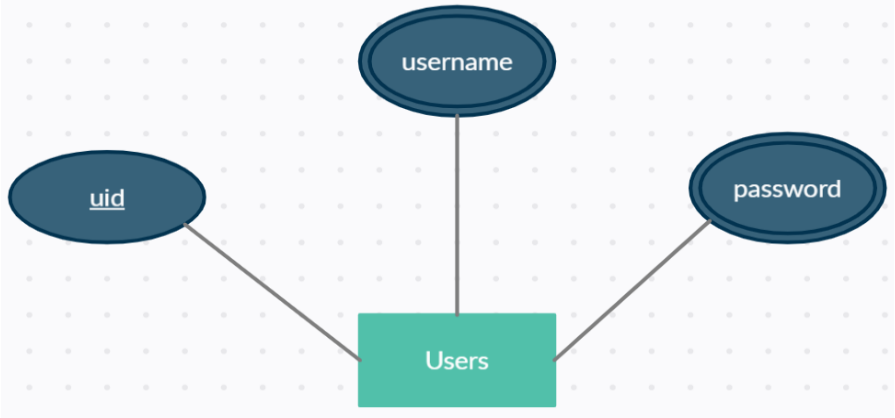
Our project's implementation is created by Python and SQLite3 and is designed with a text-based user interface. Other libraries used include blessed and pynput.

II. Database Design

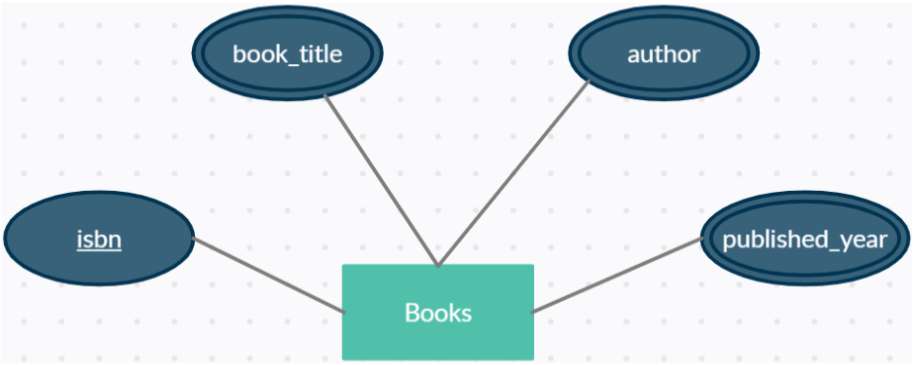
The following diagrams illustrate the architecture of our database for this project. These diagrams will be used as a reference for learning about the system's client functionality.

Entity-Relationship Diagrams:

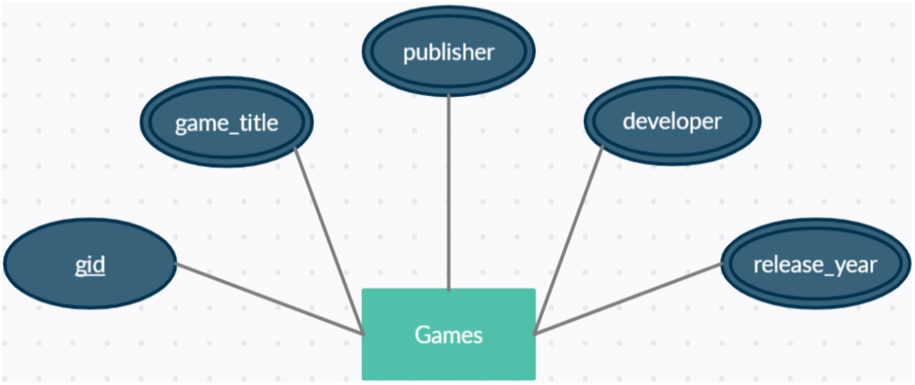
Users:



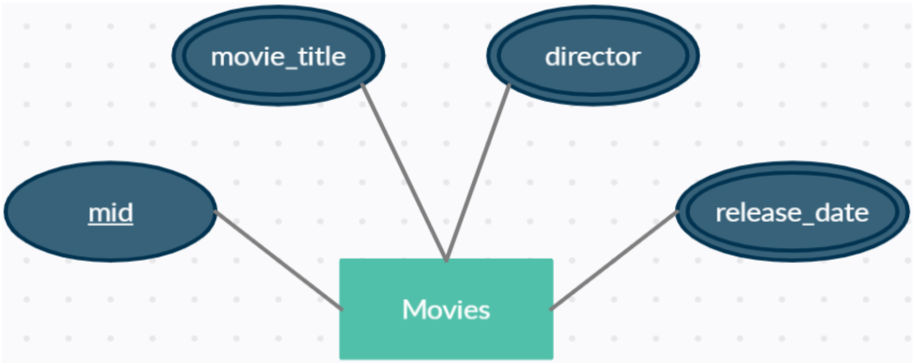
Books:



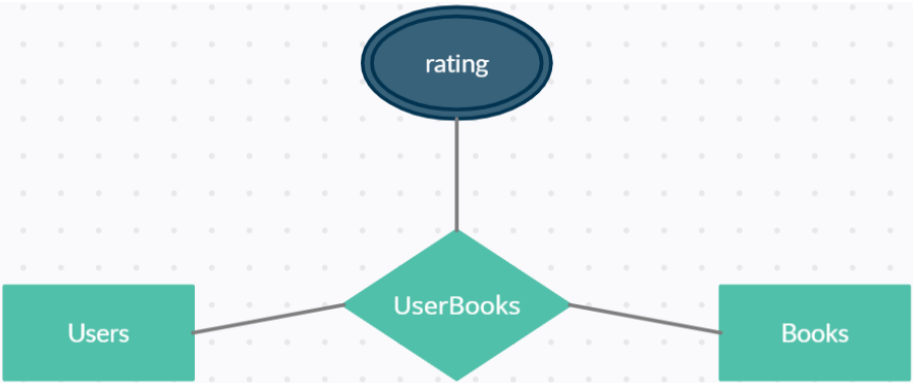
Games:



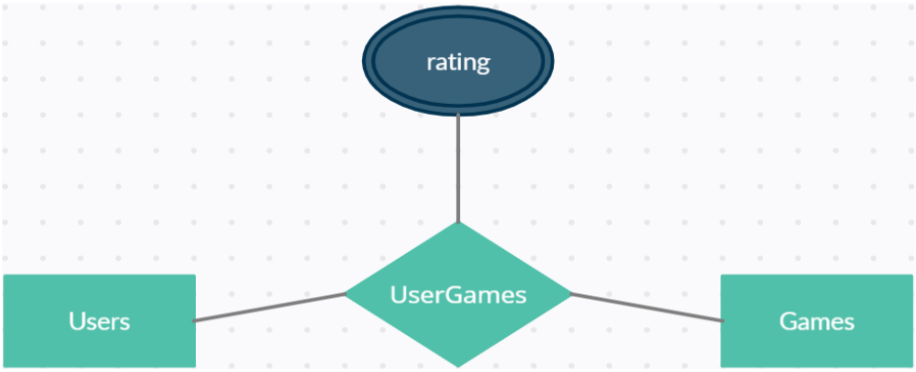
Movies:



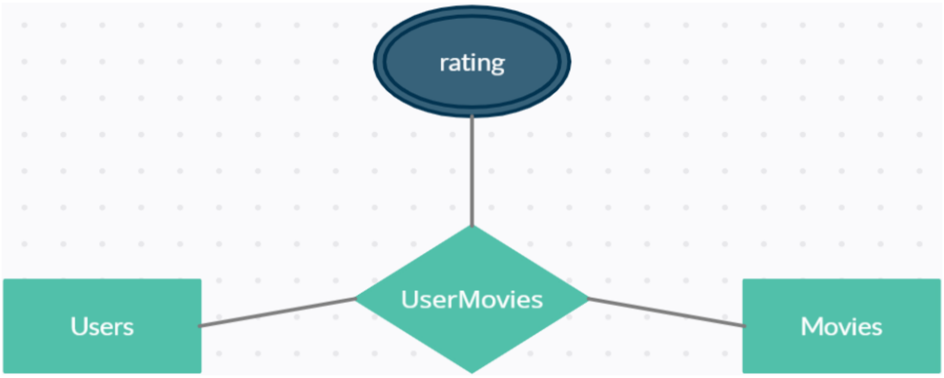
UserBooks:



UserGames:



UserMovies:



III. Client Functionality

Description:
Here describes the user interface of our system while conveying the method of use for users to interact and view the text-based interface program. In addition to this section, a detailed explanation will be provided for each type of page view and its functionality.

First, at the start up of our program, the user will be presented with a default login screen that has two options. One option is for signing up to become a user of our system and the next is to login in with an existing user credential to access the specified account's media collection.

In the case that the user is new, the user may choose their desired username and password with no constraints. The program will check to ensure there does not exist a username in the database that matches the new user's choice. Then, it will proceed to add the new user's credential to the system's record and redirect the user to the main default page, which is the *Book Page*. On the other hand, existing users will be given the option to input their existing account to gain access to their personal media library.

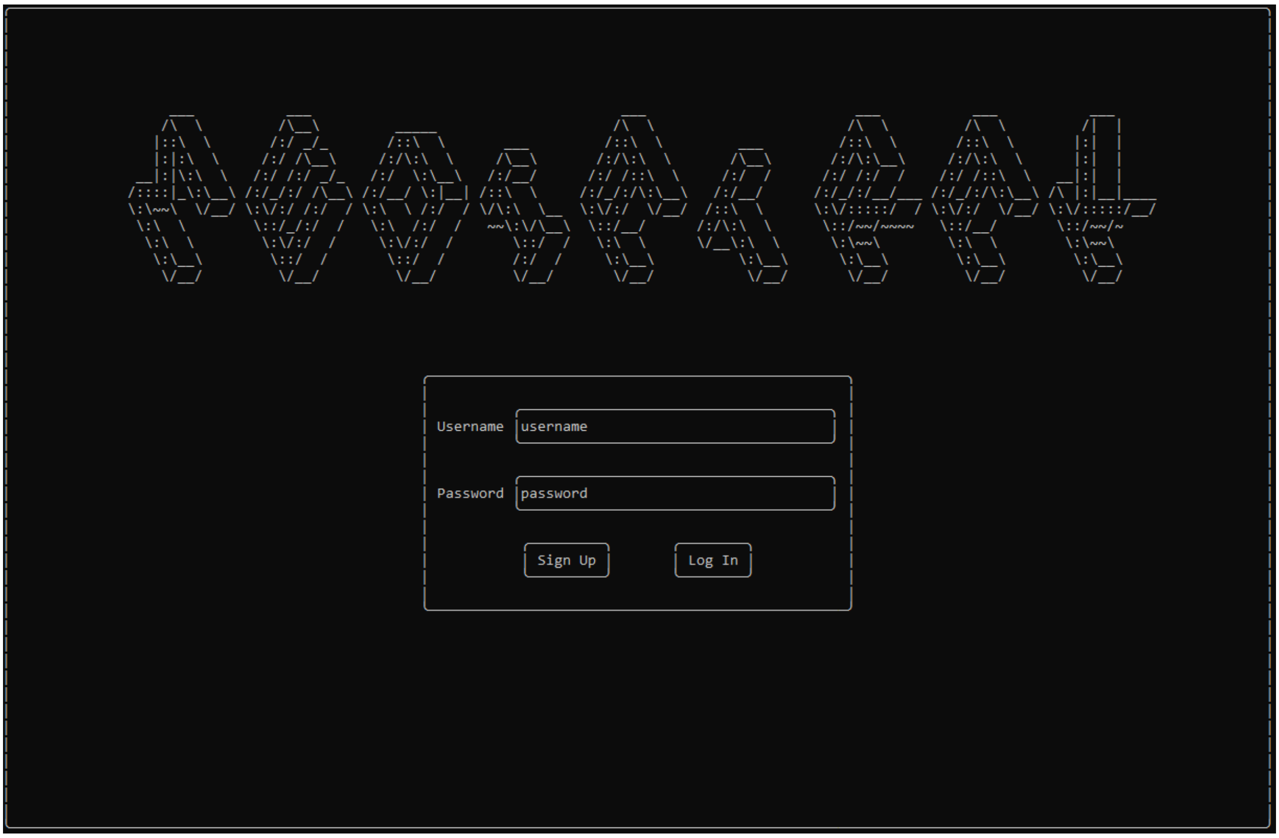
A user can view 7 types of pages:

1. Login page
2. Book Page
3. Game Page
4. Movie Page
5. UserBook Page
6. UserGame Page
7. UserMovie Page

Top Bar Navigation:
Each page view will display a navigation panel on its top section. This panel provides the user the options to navigate throughout the different pages of our system. These options include back, forward, books, games, userbooks, usernames, usermovies buttons along with a search bar.

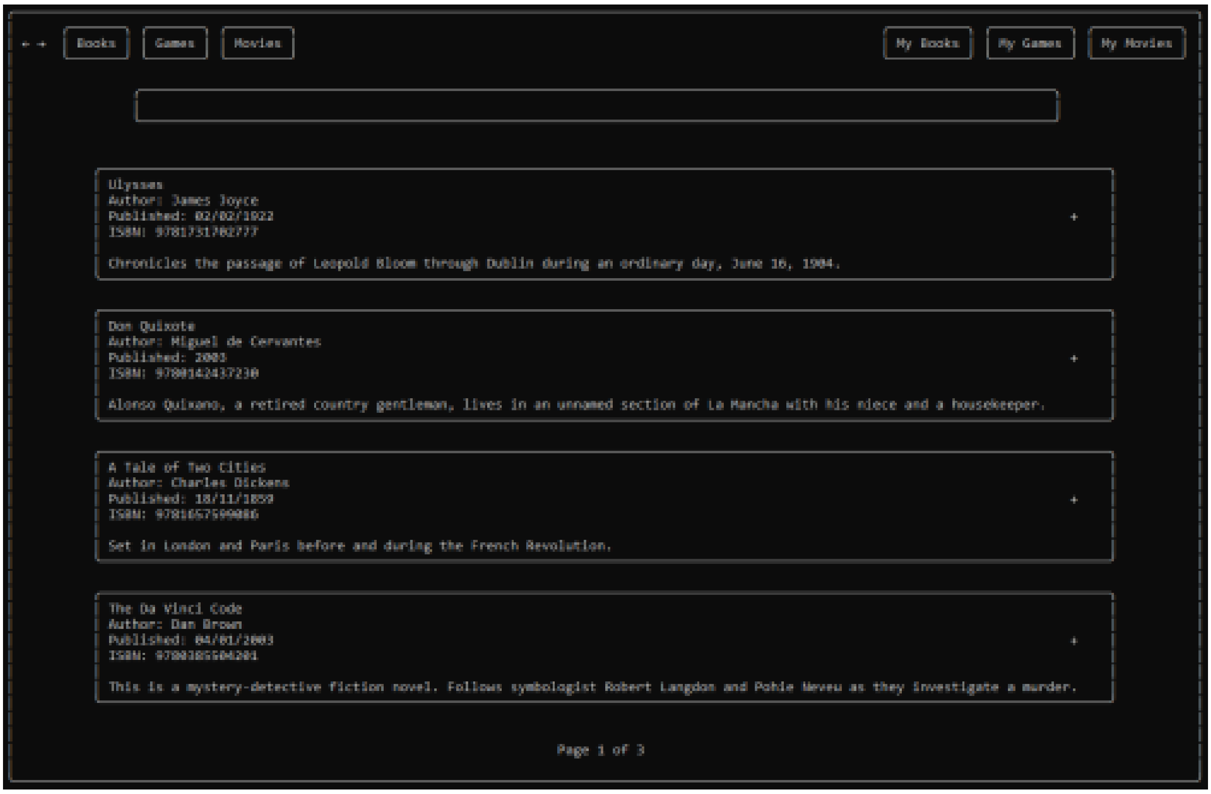
Page #1: Login Page

The Login Page will be the first page shown to new and existing users when this program is executed. Users will be welcomed by the ASCII art design of our interface and be presented with the options to sign up or login into their account. When the user logins in or creates a new account, the default main page that will be shown to the user will be the Book Page, page #2.



Page #2: Book Page

This page will consist of all existing records of books that will be displayed to the user. For each record of a book shown on this page, it will show the book's isbn, title, author, published year, a brief description. Moreover, there is a plus symbol on the right that gives the user the option to add the desired book to their collection. If a user cannot find their desired book, they could utilize the search bar at the top of the page.



```
Ulysses
Author: James Joyce
Published: 02/02/1922                                                        +
ISBN: 9781731702777

Chronicles the passage of Leopold Bloom through Dublin during an ordinary day, June 16, 1904.

Don Quixote
Author: Miguel de Cervantes
Published: 2003                                                              +
ISBN: 9780142437230

Alonso Quixano, a retired country gentleman, lives in an unnamed section of La Mancha with his niece and a housekeeper.

A Tale of Two Cities
Author: Charles Dickens
Published: 18/11/1859                                                        +
ISBN: 9781657599086

Set in London and Paris before and during the French Revolution.

The Da Vinci Code
Author: Dan Brown
Published: 04/01/2003                                                        +
ISBN: 9780385504201

This is a mystery-detective fiction novel. Follows symbologist Robert Langdon and Pohie Neveu as they investigate a murder.

                                  Page 1 of 3
```

Page #3: Game Page

This page will consist of all existing titles of games that will be displayed to the user. Each record of the game title, it will show its title, developer, publisher, release date, and a short description.

Page #4: Movie Page

This page will consist of all existing movie titles that will be displayed to the user. Each record of the movie title, it will display its title, director, release date, and a short description.

Page #5: UserBook Page

As for the UserBook Page, this is a personal page that corresponds to each unique user of the system. Each user will be given the access to select the plus symbol from the Book Page to add a record to their book collection. It will consist of all the information of the added book and an option to rate the book out of 5 stars.

Page #6: UserGame Page

Similarly to the UserBook Page, this page is where all the added games that the user chose from the Game Page will be inserted to. Users will also be given the option to rate their added games out of 5 stars.

Page #7: UserMovie Page

This page is where all the added movies that the user chose from the Movie Page will be inserted to. Users will be given the option to rate their added movies out of 5 stars.

IV. Database Tables

This section will describe which data is being stored in our database along with the relationships among each data presented in our system.
The following tables will be created in our database. Please refer to the ER diagrams for a higher-level graphical description.

Users

Users(uid: BIGSERIAL, username: CHAR(60), password: CHAR(60))
Foreign keys: none
Candidate key: username
Primary key: uid
Not null: uid, username, username, password

This table will store all the information that corresponds to each individual user of this online enterprise information system. Furthermore, this information will be used to authenticate the user to login to this web-based interface to access their media collection. The uid will be the identification number for each user to track their personal media collection.

Games

Users(gid: BIGSERIAL, game_title: CHAR(80), publisher: CHAR(60), developer: CHAR(60), release_date:CHAR(10), description:CHAR(120))
Foreign keys: none
Primary key: gid
Not null: gid, game_title, publisher, developer, release_date, description

This table will hold information associated with game entries, such as the game title, publisher, developer, and release year.

Movies

Users(mid: BIGSERIAL, movie_title:CHAR(60), director: CHAR(60), release_date: CHAR(10), description:CHAR(120))
Foreign keys: none
Primary key: mid
Not null: mid, movie_title, director, release_date, description

This table will hold information associated with movie entries, such as the movie title, director, and release date.

Books

Users(isbn:BIGSERIAL,book_title:CHAR(60),author:CHAR(60),published_year:CHAR(10), description:CHAR(120))
Foreign keys: none
Primary key: isbn
Not null: isbn, book_title, author, published_year, description

This table will hold information associated with the book entries, such as the book title, author, and publish date.

UserMovie

Users(uid:BIGSERIAL,mid:BIGSERIAL,rating:CHAR(2))
Foreign keys: (uid ->Users.uid, mid->Movies.mid)
Primary key: (mid, uid)
Not null: uid, mid, rating
This table records all added movie titles for each specified user. The field will include the uid and rating and a relation to the Movie table.

UserGame

Users(uid:BIGSERIAL,gid:BIGSERIAL,rating:CHAR(2))
Foreign keys: (uid ->Users.uid, gid->Games.gid)
Primary key: (gid, uid)
Not null: uid, gid, rating

This table records all the added game titles for each specified user. The fields will include the uid and rating. It will have a relation to the Game table.

UserBooks

Users(uid:BIGSERIAL,isbn:BIGSERIAL,rating:CHAR(2))
Foreign keys: (uid ->Users.uid, isbn->Books.isbn)
Primary key: (isbn, uid)
Not null: uid, isbn, rating

This table records all the added book titles for each specified user. The fields will include the uid and rating and will have a relation to the Books table.

V. SQL Queries

Create New Account and Login

The login page will give the user the option to sign up or login into their user accounts to gain access to the features and personal media collection. This page also takes the user input and searches the Users table to ensure there no duplicate username created for users using the following queries:

Checking Existing Users:

```
SELECT COUNT(*)
FROM users
WHERE username=username
```

Create New Users:

```
INSERT INTO users VALUES (username, password)
```

Login Existing Users:

```
SELECT COUNT(*)
FROM users
WHERE username=username AND password=password
```

Insert Queries

To insert records of books, games, and movies into our database tables. We would use the following query to insert data to their corresponding tables:

Inserting Book Information:

```
INSERT INTO books VALUES (isbn, title, author, publisher, description)
```

Inserting User Book:

```
INSERT INTO userbooks VALUES(username,isbn,rating)
```

For any media that are added to the user's media page, we must ensure that the same media is checked to ensure no duplicates are added using the following query:

Checking Existing Media:

```
SELECT COUNT(*)
FROM userbooks
WHERE username=username AND isbn=isbn
```

The above query is repeated for the other media types.

Get Queries

To retrieve a list of record for a specified media, the following query will be used to get the information of all records of books and display it to the user:

Get Media Information:

```
SELECT * FROM books
```

Get User Account's Media Information:

```
SELECT books.isbn, books.btitle, books.author, books.published,
books.bdescription, userbooks.rating
FROM books, userbooks
WHERE books.isbn=userbooks.isbn AND userbooks.username=username
```

Delete Queries

To delete a record of a selected media that a user added to their personal collection page, the following query will be used to remove a record from the table:

Deleting a Media Record:

```
DELETE FROM userbooks WHERE username=username AND isbn=isbn
```

Search Queries

To search a specific record of a selected media, the user can use the search bar to lookup their preferred media. Users will be able to search games, movies, and books based on the name of the titles. In addition, users may also search for books using its corresponding isbn. The following query will be used for the search function:

Search for Book by Title or ISBN:

```
SELECT *
FROM books
WHERE btitle LIKE search OR isbn LIKE search
```

Update Queries

The user will be able to rate each of their chosen media within their collection page. The rating scale is between 0 to 5 and will be updated using the following query:

```
UPDATE userbooks SET rating=rating WHERE username=username AND isbn=isbn
```

Count Queries

As users gradually increase their media collection by populating their personal media library. There will be a count query that tracks the total added entities for each category as follows:

```
SELECT COUNT(*)
FROM userbooks
WHERE username=username
```

VI. Database Initialization Queries

Database Creation

To create the tables for our database, the below SQL was used to initialize the database:

Users

```
CREATE TABLE users
(
   uid bigserial NOT NULL,
   username character varying NOT NULL,
   password character varying NOT NULL,
   PRIMARY KEY (uid)
);
```

Books

```
CREATE TABLE books
(
   isbn character varying NOT NULL,
   book_title character varying,
   author character varying,
   published_year integer,
   PRIMARY KEY (isbn)
);
```

Games

```
CREATE TABLE games
(
   gid bigserial NOT NULL,
   game_title character varying,
   publisher character varying,
   developer character varying,
   release_date character varying,
   PRIMARY KEY (gid)
);
```

Movies

```
CREATE TABLE movies
(
   mid bigserial NOT NULL,
   movie_title character varying,
   director character varying,
   release_date character varying,
   PRIMARY KEY (mid)
);
```

UserBooks

```
CREATE TABLE userbooks
(
   uid bigint NOT NULL,
   isbn bigint NOT NULL,
   rating integer,
   PRIMARY KEY (isbn, uid),
   FOREIGN KEY (uid) REFERENCES users (uid),
   FOREIGN KEY (isbn) REFERENCES books (isbn)
);
```

UserGames

```
CREATE TABLE usergames
(
   uid bigint NOT NULL,
   gid bigint NOT NULL,
   rating integer,
   PRIMARY KEY (gid, uid),
   FOREIGN KEY (uid) REFERENCES users (uid),
   FOREIGN KEY (gid) REFERENCES games (gid)
);
```

UserMovies

```
CREATE TABLE usermovies
(
   uid bigint NOT NULL,
   mid bigint NOT NULL,
   rating integer,
   PRIMARY KEY (mid, uid),
   FOREIGN KEY (uid) REFERENCES users (uid),
   FOREIGN KEY (mid) REFERENCES movies (mid)
);
```