

Anna Nardelli

Dr. Yu

CS 432-01

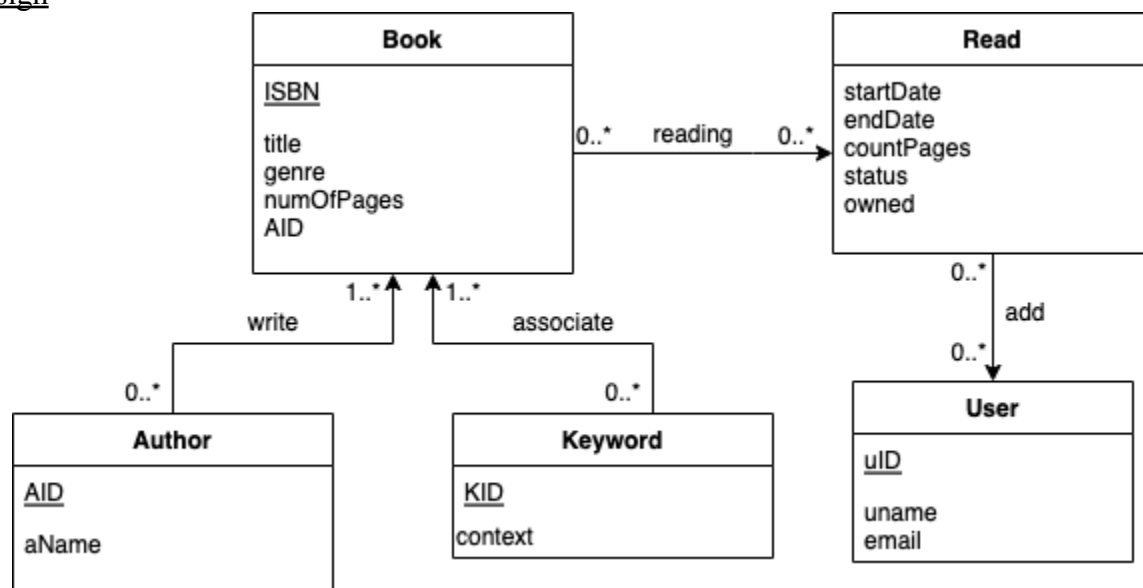
18 April 2022

## To-Be-Read Database

### Summary

This database is designed to store a user's reading list. There is a Book table that stores information about the book, including ISBN, title, and other identifying information. The Author table is separate from the Book table so that a single author can be associated with many books. The Keyword table allows a book to be associated with keywords, which aids in distinguishing connections between different books and can help with searches of the database. The User table stores information about the user and simulates the user's account. The Read table connects the Book and User tables, meaning that a user can "read" a book. This table stores information such as status, which allows a user to set the book as one that they want to read, have read, or are currently reading. The user can also record whether they own the book, when they started and finished reading, and how many pages they have read. The user can read zero to many books and a book can be read by zero to many users.

### Design



Implementation**tbrTables.sql**

```
DROP TABLE keywords;  
DROP TABLE read;  
DROP TABLE users;  
DROP TABLE author;  
DROP TABLE book;
```

```
CREATE TABLE book (  
isbn INTEGER NOT NULL,  
title VARCHAR(100) NOT NULL,  
genre VARCHAR(50) NOT NULL,  
numOfPages INTEGER NOT NULL,  
aid INTEGER NOT NULL,  
PRIMARY KEY (isbn),  
CHECK (numOfPages > 0));
```

```
CREATE TABLE author (  
aid INTEGER NOT NULL,  
aName VARCHAR(50) NOT NULL,  
PRIMARY KEY (aid));
```

```
CREATE TABLE users (  
userID INTEGER NOT NULL,  
uname VARCHAR(50) NOT NULL,  
email VARCHAR(50) NOT NULL,  
PRIMARY KEY (userId));
```

```
CREATE TABLE read (  
userID INTEGER NOT NULL,  
isbn INTEGER NOT NULL,  
startDate VARCHAR(30),  
endDate VARCHAR(30),  
countPages INTEGER,  
status VARCHAR(20) NOT NULL,  
owned CHAR NOT NULL,  
PRIMARY KEY (userId, isbn),  
FOREIGN KEY (userId) REFERENCES users(userId),  
FOREIGN KEY (isbn) REFERENCES book(isbn));
```

```
CREATE TABLE keywords (
isbn INTEGER NOT NULL,
key VARCHAR(25),
FOREIGN KEY (isbn) REFERENCES book(isbn));
```

### **tbrPopulate.sql**

```
INSERT INTO book VALUES (9787543321724, 'The Catcher in the Rye', 'bildungsroman', 277,
1);
INSERT INTO book VALUES (9781982149482, 'The Great Gatsby', 'fiction', 208, 2);
INSERT INTO book VALUES (9780062059949, 'The Seletion', 'young adult fiction', 352, 3);
INSERT INTO book VALUES (9781984822178, 'Normal People', 'fiction', 288, 4);
INSERT INTO book VALUES (9781635575569, 'A Court of Thorns and Roses', 'fantasy', 448,
5);
INSERT INTO book VALUES (9781635575583, 'A Court of Mist and Fury', 'fantasy', 656, 5);
INSERT INTO book VALUES (9780439023481, 'The Hunger Games', 'young adult fiction', 384,
6);
INSERT INTO book VALUES (9780545586177, 'Catching Fire', 'young adult fiction', 400, 6);
INSERT INTO book VALUES (9780545663267, 'Mockingjay', 'young adult fiction', 400, 6);
INSERT INTO book VALUES (9781503278196, 'Jane Eyre', 'bildungsroman', 290, 7);
INSERT INTO book VALUES (9780525559474, 'The Midnight Library', 'fiction', 304, 8);

INSERT INTO keywords VALUES (9781503278196, 'classic');
INSERT INTO keywords VALUES (9780545663267, 'ya');
INSERT INTO keywords VALUES (9781503278196, 'romance');
INSERT INTO keywords VALUES (9780545663267, 'distopian');
INSERT INTO keywords VALUES (9780545586177, 'distopian');
INSERT INTO keywords VALUES (9780439023481, 'distopian');
INSERT INTO keywords VALUES (9780439023481, 'ya');
INSERT INTO keywords VALUES (9781635575583, 'romance');
INSERT INTO keywords VALUES (9781635575569, 'romance');
INSERT INTO keywords VALUES (9781984822178, 'contemporary');
INSERT INTO keywords VALUES (9780062059949, 'romance');
INSERT INTO keywords VALUES (9781982149482, 'classic');
INSERT INTO keywords VALUES (9787543321724, 'classic');
INSERT INTO keywords VALUES (9780525559474, 'time travel');

INSERT INTO author VALUES (1, 'J.D. Salinger');
INSERT INTO author VALUES (2, 'F. Scott Fitzgerald');
INSERT INTO author VALUES (3, 'Keira Cass');
```

```

INSERT INTO author VALUES (4, 'Sally Rooney');
INSERT INTO author VALUES (5, 'Sarah J. Maas');
INSERT INTO author VALUES (6, 'Suzanne Collins');
INSERT INTO author VALUES (7, 'Charlotte Bronte');
INSERT INTO author VALUES (8, 'Matt Haig');

```

```

INSERT INTO users VALUES (1, 'Anna Nardelli', 's1164308@monmouth.edu');
INSERT INTO users VALUES (2, 'Julia Benol', 'julia@gmail.com');
INSERT INTO users VALUES (3, 'Samantha Gagnon', 'sam@aol.com');

```

```

INSERT INTO read (userId, isbn, status, owned) VALUES (1, 9781984822178, 'to be read', 0);
INSERT INTO read VALUES (1, 9780062059949, 'Jan. 2019', 'Feb. 2019', 352, 'read', 1);
INSERT INTO read VALUES (1, 9781635575583, 'Nov. 2021', 'Jan. 2022', 656, 'read', 0);
INSERT INTO read VALUES (1, 9781635575569, 'Oct. 2021', 'Nov. 2021', 448, 'read', 0);
INSERT INTO read (userId, isbn, status, owned) VALUES (1, 9781982149482, 'to be read', 1);
INSERT INTO read (userId, isbn, startDate, countPages, status, owned) VALUES (1,
9781503278196, 'April 2022', 129, 'currently reading', 1);
INSERT INTO read VALUES (2, 9780439023481, 'Aug. 2020', 'Aug. 2020', 384, 'read', 1);
INSERT INTO read VALUES (2, 9780545586177, 'Aug. 2020', 'Sept. 2020', 400, 'read', 1);
INSERT INTO read VALUES (2, 9780545663267, 'Sept. 2020', 'Sept. 2020', 400, 'read', 1);
INSERT INTO read (userId, isbn, startDate, countPages, status, owned) VALUES (2,
9781982149482, 'March 2022', 78, 'currently reading', 0);
INSERT INTO read VALUES (2, 9780062059949, 'May 2018', 'June 2018', 352, 'read', 0);
INSERT INTO read (userId, isbn, status, owned) VALUES (2, 9781503278196, 'to be read', 1);
INSERT INTO read (userId, isbn, status, owned) VALUES (2, 9781635575569, 'to be read', 0);
INSERT INTO read (userId, isbn, status, owned) VALUES (3, 9780062059949, 'to be read', 1);
INSERT INTO read VALUES (3, 9781635575569, 'Nov. 2021', 'Dec. 2021', 448, 'read', 1);
INSERT INTO read (userId, isbn, startDate, countPages, status, owned) VALUES (3,
9781635575583, 'Jan. 2022', 613, 'currently reading', 1);
INSERT INTO read VALUES (3, 9781982149482, 'Oct. 2018', 'Nov. 2018', 208, 'read', 0);
INSERT INTO read VALUES (3, 9780545663267, 'April 2017', 'April 2017', 400, 'read', 0);
INSERT INTO read (userId, isbn, status, owned) VALUES (3, 9781984822178, 'to be read', 1);

```

### **tbrQueries.sql**

```

/*Select book titles with the author Suzanne Collins
*/
SELECT book.title
FROM book, author
WHERE book.aid = author.aid AND author.aName = 'Suzanne Collins';

```

```
/*Select users that have a fantasy book on their list  
*/
```

```
SELECT DISTINCT users.uname  
FROM users, book, read  
WHERE users.userID = read.userID AND read.isbn = book.isbn AND book.genre = 'fantasy';
```

```
/*Select title and author name of books that are <250 pages  
or are young adult fiction  
*/
```

```
SELECT book.title, author.aname  
FROM book, author  
WHERE book.aid = author.aid AND (numOfPages < 250 OR genre = 'young adult fiction');
```

```
/*Select title of books that user 1 owns  
*/
```

```
SELECT book.title  
FROM users, book, read  
WHERE users.userID = read.userID AND read.isbn = book.isbn AND users.userID = 1 AND  
read.owned = 1;
```

```
/*Select authors of fiction books that were not written by Matt Haig  
*/
```

```
SELECT author.aname  
FROM book, author  
WHERE (book.aid = author.aid AND genre = 'fiction')  
MINUS  
(SELECT author.aname  
FROM book, author  
WHERE book.aid = author.aid AND author.aname = 'Matt Haig');
```

```
/*Select titles of books that user 2 has read  
*/
```

```
SELECT book.title  
FROM users, book, read  
WHERE book.isbn = read.isbn AND users.userID = read.userID AND read.userID = 2 AND  
read.status = 'read';
```

```
/*Select the book title and author with the largest number of pages  
*/
```

```
SELECT author.aname, book.title
```

```
FROM book, author
WHERE book.aid = author.aid AND numOfPages >= all
(SELECT book.numOfPages
FROM book);
```

```
/*Select book title and isbn for books that user 3 has not read
*/
SELECT book.title, book.isbn
FROM book
MINUS
(SELECT book.title, book.isbn
FROM book, users, read
WHERE book.isbn = read.isbn AND read.userID = users.userID AND read.userID = 3 AND
read.status = 'read');
```

```
/*Select start date and title of books that Anna Nardelli is currently reading
*/
SELECT read.startDate, book.title
FROM users, book, read
WHERE users.userID = read.userID AND book.isbn = read.isbn AND read.status = 'currently
reading' AND uname = 'Anna Nardelli';
```

```
/*Select authors that wrote books with keyword classic
*/
SELECT DISTINCT book.title, author.aname
FROM book, keywords, author
WHERE book.isbn = keywords.isbn AND book.aid = author.aid AND keywords.key = 'classic';
```

### Testing Conclusion

The batch files run without error. The system is effective and has been tested to ensure successful implementation of multiplicity, complete and incomplete participation, checks, and primary and foreign keys. The SQL queries are also found to be effective in identifying the correct objects that they are meant to search for. Overall, the project is correctly completed and the tests show that the implementation is concurrent with the project's description.