

## ***Wandering Words* Library Database Management System**

Kjirsten Fastabend, Allison Majors

**URL to HTML Page:**

<https://library-database-management-f2fcac56ae0f.herokuapp.com/>

## Executive Summary:

### Step 1 Changes:

- *Checkouts* entity was added to the outline to create a M:M relationship between *Books* and *Checkouts*. Foreign keys and attributes were adjusted accordingly.
- Added data types and constraints to all foreign key attributes.

The initial design had a 1:M relationship between *Books* and *Checkouts*. This restricted a book from being checked out more than once and limited the function of the database.

### Step 2 Changes:

- Intersection tables were added to the outline, along with their descriptions and removed from the ER diagram to improve readability.
- Foreign key names were rewritten to match primary keys.

In Step 2 we began progressing on our Schema diagram and building upon our outline. Intersection tables were added, and the ER diagram and outline was cleaned up for readability. Foreign keys, originally unique, were matched to primary keys.

### Step 3 Changes:

- Added pages to site for intersections tables, *BooksAuthors* and *BooksCheckouts*. *INSERT* and *SELECT* statements were added to the SQL file for these tables.

During Step 3, we focused on integrating the intersection tables fully into the design.

### Step 4 Changes:

- Null entry option added to website *INSERT* for *Employee* in *Checkout*.

Optional participation was implemented for the *EmployeeID* in *Checkouts* to simulate self-checkout.

### Step 5 Changes:

- Bug fixed for Invalid Date under *BooksCheckouts*.
- Nav Bar with toggle option added to hide forms from view.

UI updates were implemented to improve readability and a 'Checked Out' message was added for null values of Date Returned in *BooksCheckouts*.

## Project and Database Outlines:

### ***Wandering Words* Library Database Management System**

#### **Project Problem Statement:**

The library *Wandering Words* has 2,000 members. *Wandering Words* holds 5,000 books. These books may have similar or different authors. *Wandering Words* has 10 employees. *Wandering Words* needs help with keeping track of their books, their authors, checkouts, members, and employees. The database driven website will record *Checkouts of Books to Members*. This will enable visibility to which books are checked out and to which member.

#### **Entities:**

- Members: records the members belonging to *Wandering Words*
  - memberID: int, unique, not null, auto\_increment, PK
  - memberFirstName: varchar, not null
  - memberLastName: varchar, not null
  - memberEmail: varchar, not null, unique
  - Relationship: a 1:M relationship between Members and Checkouts is implemented with memberID as a FK inside of Checkouts
- Books: records the books belonging to *Wandering Words*
  - bookID: int, unique, not null, auto\_increment, PK
  - bookTitle: varchar, not null
  - genre: varchar, not null
  - numCopies: int, not null
  - Relationship: a M:M relationship between Books and Authors is implemented with a BooksAuthors intersection table that contains both a bookID as a foreign key and an authorID as a foreign key
  - Relationship: a M:M relationship between Books and Checkouts is implemented with a BooksCheckouts intersection table that

contains both a bookID as a foreign key and a checkoutID as a foreign key

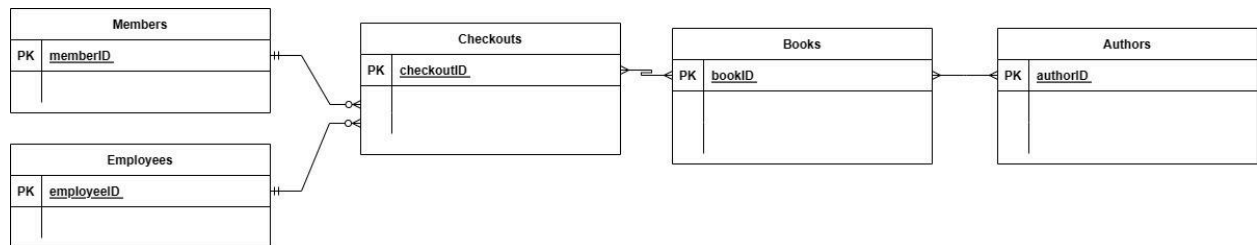
- Authors: records the authors of the books belonging to *Wandering Words*
  - authorID: int, unique, not null, auto\_increment, PK
  - authorFirstName: varchar, not null
  - authorLastName: varchar, not null
  - Relationship: a M:M relationship between Books and Authors is implemented with a BooksAuthors intersection table that contains both a bookID as a foreign key and an authorID as a foreign key
- Checkouts: records the event of a book/books being checked out to a member
  - checkoutID: int, unique, not null, auto\_increment, PK
  - dateCheckedOut: date, not null
  - dateDue: date, not null
  - memberID: FK, int, unique, not null, auto\_increment
  - employeeID: FK, int, unique, auto\_increment
  - Relationship: a M:1 relationship between Checkouts and Members is implemented with memberID as a FK inside of Checkouts
  - Relationship: a M:1 relationship between Checkouts and Employees is implemented with employeeID as a foreign key inside of Checkouts
  - Relationship: a M:M relationship between Books and Checkouts is implemented with a BooksCheckouts intersection table that contains both a bookID as a foreign key and a checkoutID as a foreign key
- Employees: Active employees at *Wandering Words*, records who is responsible for Checkouts transactions
  - employeeID: int, unique, not null, auto\_increment, PK
  - employeeFirstName: varchar, not null
  - employeeLastName: varchar, not null
  - employeeEmail: varchar, not null, unique

- Relationship: a 1:M relationship between Employees and Checkouts is implemented with employeeID as a foreign key inside of Checkouts

### Intersection tables:

- BooksAuthors: implements the M:M relationship between Books and Authors
  - booksAuthorsID: int, unique, not null, auto\_increment, PK
  - bookID: FK, int, unique, not null, auto\_increment
  - authorID: FK, int, unique, not null, auto\_increment
- BooksCheckouts: implements the M:M relationship between Books and Checkouts
  - booksCheckoutsID: int, unique, not null, auto\_increment, PK
  - bookID: FK, int, unique, not null, auto\_increment
  - checkoutID: FK, int, unique, not null, auto\_increment
  - dateReturned: date

### ER Diagram:



```

    erDiagram
        Members ||--o{ Checkouts : "has"
        Employees ||--o{ Checkouts : "has"
        Checkouts ||--|| BooksCheckouts : "has"
        Books ||--|| BooksCheckouts : "has"
        Books ||--|| BooksAuthors : "has"
        Authors ||--|| BooksAuthors : "has"

        Members {
            INT memberID PK
            VARCHAR(255) memberFirstName
            VARCHAR(255) memberLastName
            VARCHAR(255) memberEmail
        }
        Employees {
            INT employeeID PK
            VARCHAR(255) employeeFirstName
            VARCHAR(255) employeeLastName
            VARCHAR(255) employeeEmail
        }
        Checkouts {
            INT checkoutID PK
            INT memberID FK
            INT employeeID FK
            DATE dateCheckedOut
            DATE dateDue
        }
        Books {
            INT bookID PK
            VARCHAR(255) bookTitle
            VARCHAR(45) genre
            INT numCopies
        }
        Authors {
            INT authorID PK
            VARCHAR(255) authorFirstName
            VARCHAR(255) authorLastName
        }
        BooksCheckouts {
            INT booksCheckoutsID PK
            INT checkoutID FK
            INT bookID FK
            DATE dateReturned
        }
        BooksAuthors {
            INT booksAuthorsID PK
            INT bookID FK
            INT authorID FK
        }
  
```

The diagram illustrates a library database schema with the following tables and attributes:

- Members**: memberID (PK), memberFirstName, memberLastName, memberEmail.
- Employees**: employeeID (PK), employeeFirstName, employeeLastName, employeeEmail.
- Checkouts**: checkoutID (PK), memberID (FK), employeeID (FK), dateCheckedOut, dateDue.
- Books**: bookID (PK), bookTitle, genre, numCopies.
- Authors**: authorID (PK), authorFirstName, authorLastName.
- BooksCheckouts**: booksCheckoutsID (PK), checkoutID (FK), bookID (FK), dateReturned.
- BooksAuthors**: booksAuthorsID (PK), bookID (FK), authorID (FK).

Relationships are shown as follows:

- Members** to **Checkouts**: One-to-many relationship.
- Employees** to **Checkouts**: One-to-many relationship.
- Checkouts** to **BooksCheckouts**: One-to-many relationship.
- Books** to **BooksCheckouts**: One-to-many relationship.
- Books** to **BooksAuthors**: One-to-many relationship.
- Authors** to **BooksAuthors**: One-to-many relationship.

Indexes are indicated for the primary keys and foreign keys in each table.

Members:

### Books:

bookID	bookTitle	genre	numCopies
1	Harry Potter and the Chamber of Secrets	fantasy	5

2	Harry Potter and the Prisoner of Azkaban	fantasy	5
3	Sherlock Holmes	mystery	5
4	Pride and Prejudice	historical fiction	5
5	The Talisman	fantasy	5

#### Authors:

authorID	authorFirstName	authorLastName
1	J.K.	Rowling
2	Jane	Austen
3	Conan	Doyle
4	Stephen	King
5	Peter	Straub

#### BooksAuthors:

booksAuthorsID	bookID (FK)	authorID(FK)
1	1	1
2	2	1
3	3	3
4	4	2
5	5	4
6	5	5

#### Checkouts:

checkoutID	memberID (FK)	employeeID (FK)	dateCheckedOut	dateDue
1	3	1	04/09/24	04/29/24
2	3	5	03/05/23	03/25/23
3	2	4	11/02/23	11/23/23

4	1	4	05/05/22	05/25/22
5	4	2	12/06/23	12/26/23

#### BooksCheckouts:

booksCheckoutsID	bookID (FK)	checkoutID (FK)	dateReturned
1	2	1	NULL
2	2	2	03/20/23
3	3	4	05/22/22
4	5	4	05/10/22
5	4	3	11/20/23

#### Employees:

employeeID	employeeFirstName	employeeLastName	employeeEmail
1	Haley	Smith	haleysmith@gmail.com
2	John	Martin	johnmartin@gmail.com
3	Sue	Sylvester	suesylvester@gmail.com
4	Brad	Pitt	bradpitt@gmail.com
5	Annie	Hall	anniehall@gmail.com



# UI Screen Shots with Informative Titles:

INSERT/DISPLAY/UPDATE/DELETE Checkouts page

(Nullable relationship with Employee ID)

(Delete from a M:N (cascade removes a Checkout from BooksCheckouts as well))

Not Secure classwork.engr.oregonstate.edu:9365/#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

Checkouts

Checkout ID	Member ID	Employee ID	Date Checked Out	Date Due	Delete
1	Eddie Murphy	Haley Smith	04-09-2024	04-29-2024	Delete
2	Eddie Murphy	Annie Hall	03-05-2023	03-25-2023	Delete
3	John Cena	Brad Pitt	11-02-2023	11-23-2023	Delete
4	Taylor Swift	Brad Pitt	05-05-2022	05-25-2022	Delete
5	Hannah Montana	John Martin	12-06-2023	12-26-2023	Delete
6	Hannah Montana	Self Checkout (NULL)	12-02-2023	12-22-2023	Delete

Add Checkout

Update Checkout

Show All Checked Out Books

Checked Out Books

Checkout ID	Book Title	Date Returned
1	Harry Potter and the Prisoner of Azkaban	Checked Out
2	Harry Potter and the Prisoner of Azkaban	03-20-2023
3	Pride and Prejudice	11-20-2023
4	Sherlock Holmes	06-12-2024
4	The Talisman	06-28-2024

Add a New Checkout

To add a new checkout, please enter its information below and click 'Submit'!

## INSERT/DISPLAY/UPDATE/DELETE Checkouts page (cont.)

⏮ ⏭ ↻

⚠ Not Secure classwork.engr.oregonstate.edu:9365/#

## Add a New Checkout

To add a new checkout, please enter its information below and click 'Submit'!

**Member:**

Select a Member ▾

**Employee:**

Select an Employee ▾

**Date checked out:**

mm/dd/yyyy 📅

**Date due:**

mm/dd/yyyy 📅

Submit

## Update a Checkout

To update a checkout, please fill in the below fields and click 'Submit'!

**Checkout:**

Select an Checkout ▾

**Member:**

Select a Member ▾

**Employee:**

Select an Employee ▾

**Date checked out:**

mm/dd/yyyy 📅

**Date due:**

mm/dd/yyyy 📅

Submit

## INSERT/DISPLAY Authors page

Not Secure classwork.engr.oregonstate.edu:9365/authors#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

### Authors

Author ID	First Name	Last Name
1	J.K.	Rowling
2	Jane	Austen
3	Conan	Doyle
4	Stephen	King
5	Peter	Straub

Add Author

### Add a new Author

To add a new author, please enter its information below and click 'Submit'!

**First name:**

**Last name:**

Submit

## INSERT/DISPLAY Books page

Not Secure classwork.engr.oregonstate.edu:9365/books#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

### Books

Book ID	Book Title	Genre	Number of Copies
1	Harry Potter and the Chamber of Secrets	fantasy	5
2	Harry Potter and the Prisoner of Azkaban	fantasy	5
3	Sherlock Holmes	mystery	5
4	Pride and Prejudice	mystery	5
5	The Talisman	fantasy	5

Add Book

### Add a new Book

To add a new book, please enter its information below and click 'Submit'!

**Book title:**

**Genre:**

**Number of copies:**

Submit

## INSERT/DISPLAY Members page

Not Secure classwork.engr.oregonstate.edu:9365/members#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

### Members

Member ID	First Name	Last Name	Email
1	Taylor	Swift	swiftie101@gmail.com
2	John	Cena	biker23@gmail.com
3	Eddie	Murphy	edmurphy@yahoo.com
4	Hannah	Montana	cowgirl3@yahoo.com
5	Miley	Cyrus	singsingsing@gmail.com

Add Member

### Add a new Member

To add a new member, please enter its information below and click 'Submit'!

**First name:**

**Last name:**

**Email:**

Submit

## INSERT/DISPLAY Employees page

← → ↻ ⚠ Not Secure classwork.engr.oregonstate.edu:9365/employees#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

### Employees

Employee ID	First Name	Last Name	Email
1	Hailey	Smith	haileysmith@gmail.com
2	John	Martin	johnmartin@gmail.com
3	Sue	Sylvester	suesylvester@gmail.com
4	Brad	Pitt	bradpitt@gmail.com
5	Annie	Hall	anniehall@gmail.com

Add Employee

### Add a new Employee

To add a new employee, please enter their information below and click 'Submit'!

**First name:**

**Last name:**

**Email:**

Submit

## INSERT/DISPLAY BooksAuthors page

← → ↺

⚠ Not Secure classwork.engr.oregonstate.edu:9365/booksAuthors#

CheckoutsAuthorsBooksMembersEmployeesBooks with AuthorsBooks in Checkouts

### Books with Authors

Author Book ID	Book ID	Author ID
1	1	1
2	2	1
3	3	3
4	4	2
5	5	4
6	5	5

Pair Book to Existing Author

### Add a New Book Author Connection

To add a book author connection, please enter its information below and click 'Submit'!

**Book:**

Select a Book

**Author:**

Select an Author

Submit

INSERT/DISPLAY/UPDATE/DELETE BooksCheckouts page

(Delete from a M:N)

(Update of a M:N)

⚠ Not Secure classwork.engr.oregonstate.edu:9365/booksCheckouts#

Checkouts

Authors

Books

Members

Employees

Books with Authors

Books in Checkouts

Books in Checkouts

Book Checkout ID	Checkout ID	Book ID	Date Returned	Delete
1	1	2	Checked Out	Delete
2	2	2	03-20-2023	Delete
3	4	3	06-12-2024	Delete
4	4	5	06-28-2024	Delete
5	3	4	11-20-2023	Delete

Add Book to Checkout

Update Book in Checkout

Update Book Return Date

Add a Book to a Checkout

To add a book to a checkout, please enter its information below and click 'Submit'!

Book:

Select a Book

Checkout:

Select a Checkout

Date Returned:

mm/dd/yyyy

Submit

Updating A Book in a Checkout

To update a book in a Checkout, please select the Book Checkout ID and fill in the corresponding forms, then click 'Submit'!

Book Checkout ID:

## INSERT/DISPLAY/UPDATE/DELETE BooksCheckouts page (cont.)

### (Update of a M:N)

[←](#) [→](#) [↻](#) ⚠ Not Secure classwork.engr.oregonstate.edu:9365/booksCheckouts#

[Add Book to Checkout](#) [Update Book in Checkout](#) [Update Book Return Date](#)

### Add a Book to a Checkout

To add a book to a checkout, please enter its information below and click 'Submit'!

**Book:**

Select a Book

**Checkout:**

Select a Checkout

**Date Returned:**

mm/dd/yyyy

### Updating A Book in a Checkout

To update a book in a Checkout, please select the Book Checkout ID and fill in the corresponding forms, then click 'Submit'!

**Book Checkout ID:**

Select a Checkout

**Book:**

Select a Book

### Updating A Return Date in a Checkout

To update the return date, please select the Book Checkout ID and fill in the corresponding forms, then click 'Submit'!

**Book Checkout ID:**

Select a Checkout

**Date Returned:**

mm/dd/yyyy