**Carmen Municipal College**

**A.Y.2024-2025**

**CC105-Books Management System**

**Midterm Project**

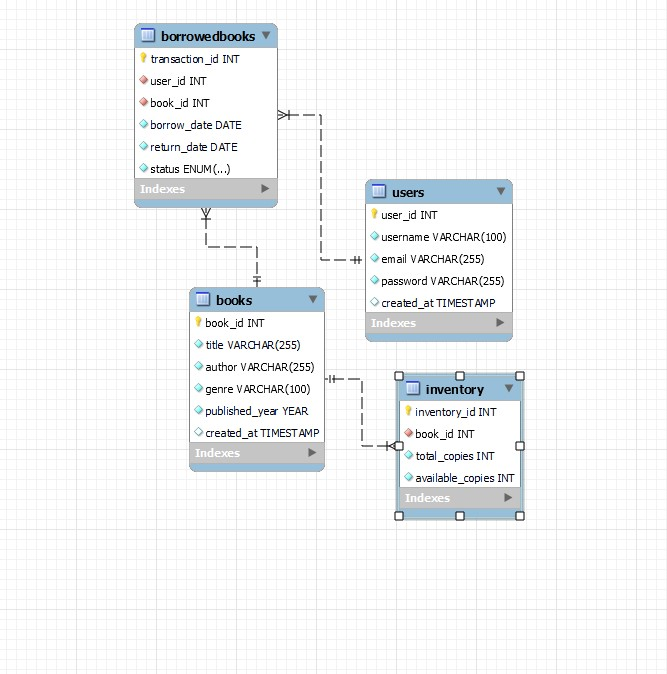
**John Louie Romero**

**Junie H. Saludes**

1. **Introduction**

A **Book Management System** is a software application designed to efficiently manage books, users, and borrowing transactions. It provides a structured way to store book details, track inventory, and monitor borrowing activities. This system is commonly used in **libraries, bookstores, and educational institutions** to streamline book lending and management.

1. **Entity Relationship Diagram**



**III.Table (Stores Book Details)**

**Purpose:**

Stores information about books available in the system.

**Columns:**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| book\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each book |
| title | VARCHAR(255) | NOT NULL, CHECK (LENGTH(title) > 0) | Title of the book |
| author | VARCHAR(255) | NOT NULL, CHECK (LENGTH(author) > 0) | Name of the book’s author |
| genre | VARCHAR(100) | NOT NULL, CHECK (LENGTH(genre) > 0) | Book category or genre (e.g., Fiction, Science) |
| published\_year | YEAR | NOT NULL | Year the book was published |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp when the book entry was added |

**Constraints:**

* **CHECK Constraints**: Ensures title, author, and genre are not empty.
* **Primary Key (book\_id)**: Ensures each book has a unique identifier.

**2. Users Table (Stores User Details)**

**Purpose:**

Stores details about users who can borrow books.

**Columns:**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| user\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each user |
| username | VARCHAR(100) | UNIQUE, NOT NULL, CHECK (LENGTH(username) > 0) | Unique username for the user |
| email | VARCHAR(255) | UNIQUE, NOT NULL, CHECK (LENGTH(email) > 0) | User’s email (must be unique) |
| password | VARCHAR(255) | NOT NULL, CHECK (LENGTH(password) > 0) | Hashed password for authentication |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp when the user was registered |

**Constraints:**

* **UNIQUE (username and email)**: Ensures no duplicate users.
* **CHECK Constraints**: Ensures username, email, and password are not empty.
* **Primary Key (user\_id)**: Ensures each user has a unique identifier.

**3. Inventory Table (Stores Book Copies)**

**Purpose:**

Tracks the number of available and total copies of each book.

**Columns:**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| inventory\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for inventory records |
| book\_id | INT | FOREIGN KEY | References Books(book\_id) |
| total\_copies | INT | NOT NULL, CHECK (total\_copies >= 0) | Total copies of the book in the system |
| available\_copies | INT | NOT NULL, CHECK (available\_copies >= 0) | Number of copies available for borrowing |

**Constraints:**

* **CHECK Constraints**: Ensures total\_copies and available\_copies are not negative.
* **Foreign Key (book\_id)**: References Books(book\_id), ensuring only valid books exist in inventory.
* **ON DELETE CASCADE**: If a book is deleted, its inventory record is also removed.

**4. BorrowedBooks Table (Stores Borrowing Transactions)**

**Purpose:**

Records the transactions of users borrowing and returning books.

**Columns:**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| transaction\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each transaction |
| user\_id | INT | FOREIGN KEY | Links to the Users table |
| book\_id | INT | FOREIGN KEY | Links to the Books table |
| borrow\_date | DATE | NOT NULL | Date when the book was borrowed |
| return\_date | DATE | NOT NULL | Date when the book should be returned |
| status | ENUM('borrowed', 'returned') | DEFAULT 'borrowed' | Status of the transaction |

CREATE DATABASE IF NOT EXISTS BookManagementDB;

USE BookManagementDB;

CREATE TABLE IF NOT EXISTS Books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

genre VARCHAR(100) NOT NULL,

published\_year YEAR NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT chk\_title CHECK (LENGTH(title) > 0),

CONSTRAINT chk\_author CHECK (LENGTH(author) > 0),

CONSTRAINT chk\_genre CHECK (LENGTH(genre) > 0)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(100) UNIQUE NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT chk\_username CHECK (LENGTH(username) > 0),

CONSTRAINT chk\_email CHECK (LENGTH(email) > 0),

CONSTRAINT chk\_password CHECK (LENGTH(password) > 0)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS Inventory (

inventory\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT NOT NULL,

total\_copies INT NOT NULL CHECK (total\_copies >= 0),

available\_copies INT NOT NULL CHECK (available\_copies >= 0),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id) ON DELETE CASCADE

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS BorrowedBooks (

transaction\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

book\_id INT NOT NULL,

borrow\_date DATE NOT NULL,

return\_date DATE NOT NULL,

status ENUM('borrowed', 'returned') NOT NULL DEFAULT 'borrowed',

FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE,

FOREIGN KEY (book\_id) REFERENCES Books(book\_id) ON DELETE CASCADE

) ENGINE=InnoDB;

INSERT INTO Users (username, email, password) VALUES ('Alice', 'alice@example.com', 'password123');

INSERT INTO Books (title, author, genre, published\_year) VALUES ('The Great Gatsby', 'F. Scott Fitzgerald', 'Classic', 1925);

INSERT INTO Inventory (book\_id, total\_copies, available\_copies) VALUES (1, 10, 10);

INSERT INTO BorrowedBooks (user\_id, book\_id, borrow\_date, return\_date) VALUES (1, 1, '2024-03-01', '2024-03-15');

SELECT

Books.title,

Books.author,

Inventory.total\_copies,

Inventory.available\_copies

FROM Books

JOIN Inventory ON Books.book\_id = Inventory.book\_id;

SELECT

BorrowedBooks.transaction\_id,

Users.username,

Books.title,

BorrowedBooks.borrow\_date,

BorrowedBooks.return\_date,

BorrowedBooks.status

FROM BorrowedBooks

JOIN Users ON BorrowedBooks.user\_id = Users.user\_id

JOIN Books ON BorrowedBooks.book\_id = Books.book\_id;