Logical Database Schema

(BOOKed)

Entity-Relationship Mapping

- **Book**: Represents a book in the library's inventory.
- **User:** Represents any user who can borrow or access the library's services.
- **Regular User**: A subtype of User, representing a general library member.
- **Librarian:** A subtype of User, representing a library staff member.
- **Borrowing**: Tracks the borrowing history of books, including the user, book, borrowing date, and return date.
- **Reservation:** Tracks reservations, including the user, book, and reservation date.
- **Review:** Represents reviews of books, including review text, rating, and the reviewer.
- **Account Management:** Tracks user profile updates, including edit dates.

Mapping Approach

One-to-Many (1:N) Relationships

- **Book** ↔ **Borrowing:** A book can have many borrowing records (multiple users can borrow the same book at different times).
- **Book** ↔ **Reservation**: A book can have many reservations (many users can reserve a book).
- **Book** ↔ **Review:** A book can have multiple reviews from different users.
- **User** ↔ **Borrowing:** A user can borrow multiple books.
- **User** ↔ **Reservation**: A user can reserve multiple books.
- **User** ↔ **Review:** A user can write multiple reviews for different books

One-to-One (1:1) Relationships

- **User** ↔ **Regular User or Librarian:** A user is either a regular user or a librarian. This is modeled by having separate Regular_User and Librarian tables, both referencing User. The relationship is enforced with foreign key constraints.

Many-to-Many (M:N) Relationships

- **User** ↔ **Book (via Borrowing):** A user can borrow multiple books, and a book can be borrowed by multiple users, implemented through the Borrowing table.
- **User** ↔ **Book (via Reservation):** A user can reserve multiple books, and a book can be reserved by multiple users, implemented through the Reservation table.

Table Structure

- **Book:** Stores information about the book, such as title, author, ISBN, and available copies.
- **User:** Stores user information including name and email, which is unique for each user.
- **Regular User and Librarian:** These tables store subtype-specific information for regular users and librarians respectively, both linked to the User table.
- **Borrowing:** Stores borrowing transactions, including User_ID, Book_ID, borrowing date, and optional return date.
- **Reservation:** Stores information about user reservations, including User_ID, Book_ID, and reservation date.
- **Review:** Stores reviews of books, with fields for User_ID, Book_ID, review content, rating, and the date of the review.
- **Account Management:** Stores records of profile edits, such as the date of the last profile update for each user.

Foreign Key Constraints

Used to enforce relationships between the different entities:

- **Borrowing Table:**User_ID references the User table.
    - Book_ID references the Book table.

- **Reservation Table:**
    - User_ID references the User table.
    - Book_ID references the Book table.

- **Review Table:**
    - User_ID references the User table.
    - Book_ID references the Book table.

- **Account Management Table:**
    - User_ID references the User table.

The **ON DELETE CASCADE** clause ensures that when a user or book is deleted, all related borrowing, reservation, review, and account management records are automatically removed. This helps maintain referential integrity and prevents orphaned records.

- To demonstrate the system's functionality, this is an example of how data is stored and retrieved:

User_ID	Name	Email
1	Alice Johnson	alice.johnson@example.com
2	Bob Smith	bob.smith@example.com

Book_ID	Title	Author	ISBN	Available_Copies
1	The Great Gatsby	F. Scott Fitzgerald	9780743273565	5
2	1984	George Orwell	9780451524935	3

Borrowing_ID	User_ID	Book_ID	Borrow_Date	Return_Date	Status
1	1	1	2024-09-10	2024-09-20	Borrowed

Reservation_ID	User_ID	Book_ID	Reservation_Date	Status
1	2	2	2024-09-15	Active