TABLES

Books

Authors

Publishers

Members

Loans

BookAuthors

1. Books

```
CREATE TABLE Books (
    BookID INT AUTO_INCREMENT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    PublisherID INT,
    ISBN VARCHAR(13) UNIQUE NOT NULL,
    YearPublished YEAR CHECK (YearPublished >= 1450 AND YearPublished <= YEAR(CURDATE())),
    Genre VARCHAR(50),
    CopiesAvailable INT DEFAULT 1 CHECK (CopiesAvailable >= 0),
    FOREIGN KEY (PublisherID) REFERENCES Publishers(PublisherID)
);
```

2. Authors

```
CREATE TABLE Authors (
    AuthorID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100) NOT NULL,
    DateOfBirth DATE
);
```

3. Publishers

```
CREATE TABLE Publishers (
    PublisherID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL UNIQUE,
    Address VARCHAR(255),
    Phone VARCHAR(20),
```

```
    Email VARCHAR(100) UNIQUE
);
4. Members
CREATE TABLE Members (
    MemberID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Phone VARCHAR(20),
    Email VARCHAR(100) UNIQUE,
    DateOfMembership DATE NOT NULL,
    MembershipType ENUM('Regular', 'Premium') DEFAULT 'Regular',
    CHECK (MembershipType IN ('Regular', 'Premium'))
);
5. Loans
CREATE TABLE Loans (
    LoanID INT AUTO_INCREMENT PRIMARY KEY,
    BookID INT NOT NULL,
    MemberID INT NOT NULL,
    LoanDate DATE NOT NULL,
    ReturnDate DATE,
    DueDate DATE NOT NULL,
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    CHECK (LoanDate <= DueDate),
    CHECK (ReturnDate IS NULL OR ReturnDate >= LoanDate)
);
6. BookAuthors
CREATE TABLE BookAuthors (
    BookID INT NOT NULL,
    AuthorID INT NOT NULL,
```

```
    PRIMARY KEY (BookID, AuthorID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);
```

Explanation of Constraints

NOT NULL: Ensures that the field must have a value.

UNIQUE: Ensures that all values in the field are unique across the table.

CHECK: Ensures that the field value meets a specific condition.

PRIMARY KEY: Uniquely identifies each record in the table.

FOREIGN KEY: Ensures referential integrity by linking to the primary key of another table.

Relationships

Books to Publishers: Each book may have one publisher (Many-to-One).

Books to Authors: Many-to-Many relationship (handled by the BookAuthors table).

Members to Loans: Each member can have multiple loans (One-to-Many).

Books to Loans: Each book can be loaned multiple times (One-to-Many).

This schema design should handle the basic functionality of a library system, ensuring data integrity and establishing necessary relationships between entities.