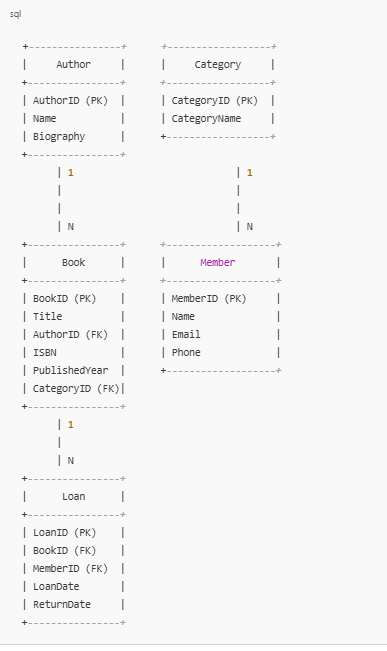
**ERD Diagram**



**Book**

* 1. **BookID** (Primary Key)
  2. Title
  3. AuthorID (Foreign Key)
  4. ISBN
  5. PublishedYear
  6. CategoryID (Foreign Key)

**Author**

* 1. **AuthorID** (Primary Key)
  2. Name
  3. Biography

**Category**

* 1. **CategoryID** (Primary Key)
  2. CategoryName

**Member**

* 1. **MemberID** (Primary Key)
  2. Name
  3. Email
  4. Phone

**Loan**

* 1. **LoanID** (Primary Key)
  2. BookID (Foreign Key)
  3. MemberID (Foreign Key)
  4. LoanDate
  5. ReturnDate

### Relationships

* **One-to-One Relationship**: Each author can have a biography (1:1), but we will not strictly enforce it in this case, allowing authors to have zero or one biography.
* **One-to-Many Relationship**:
  + One author can write many books (1).
  + One category can include many books (1).
  + One member can have multiple loans (1).
* **Many-to-Many Relationship**: This is represented by the Loan entity. A book can be loaned to many members over time, and a member can loan many books.

**-- Create the Author table**

CREATE TABLE Author (

AuthorID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Biography TEXT

);

**-- Create the Category table**

CREATE TABLE Category (

CategoryID INT PRIMARY KEY AUTO\_INCREMENT,

CategoryName VARCHAR(100) NOT NULL

);

**-- Create the Book table**

CREATE TABLE Book (

BookID INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(200) NOT NULL,

AuthorID INT,

ISBN VARCHAR(20) UNIQUE,

PublishedYear YEAR,

CategoryID INT,

FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),

FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID)

);

**-- Create the Member table**

CREATE TABLE Member (

MemberID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Phone VARCHAR(15)

);

**-- Create the Loan table**

CREATE TABLE Loan (

LoanID INT PRIMARY KEY AUTO\_INCREMENT,

BookID INT,

MemberID INT,

LoanDate DATE NOT NULL,

ReturnDate DATE,

FOREIGN KEY (BookID) REFERENCES Book(BookID),

FOREIGN KEY (MemberID) REFERENCES Member(MemberID)

);