Assignment 2: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

Tables and Fields for Library System

Books

book\_id (Primary Key, INT, NOT NULL, AUTO\_INCREMENT)

title (VARCHAR(255), NOT NULL)

author\_id (Foreign Key, INT, NOT NULL)

genre\_id (Foreign Key, INT, NOT NULL)

isbn (VARCHAR(13), UNIQUE, NOT NULL)

publish\_date (DATE, NOT NULL)

copies\_available (INT, NOT NULL, CHECK (copies\_available >= 0))

Authors

author\_id (Primary Key, INT, NOT NULL, AUTO\_INCREMENT)

first\_name (VARCHAR(50), NOT NULL)

last\_name (VARCHAR(50), NOT NULL)

Genres

genre\_id (Primary Key, INT, NOT NULL, AUTO\_INCREMENT)

genre\_name (VARCHAR(50), UNIQUE, NOT NULL)

Members

member\_id (Primary Key, INT, NOT NULL, AUTO\_INCREMENT)

first\_name (VARCHAR(50), NOT NULL)

last\_name (VARCHAR(50), NOT NULL)

email (VARCHAR(100), UNIQUE, NOT NULL)

phone (VARCHAR(15), UNIQUE)

membership\_date (DATE, NOT NULL)

Loans

loan\_id (Primary Key, INT, NOT NULL, AUTO\_INCREMENT)

book\_id (Foreign Key, INT, NOT NULL)

member\_id (Foreign Key, INT, NOT NULL)

loan\_date (DATE, NOT NULL)

return\_date (DATE)

due\_date (DATE, NOT NULL, CHECK (due\_date > loan\_date))

Relationships and Constraints

Books Table

book\_id: Primary Key, uniquely identifies each book.

author\_id: Foreign Key references Authors.author\_id.

genre\_id: Foreign Key references Genres.genre\_id.

isbn: Unique identifier for each book.

copies\_available: Must be a non-negative number.

Authors Table

author\_id: Primary Key, uniquely identifies each author.

Genres Table

genre\_id: Primary Key, uniquely identifies each genre.

genre\_name: Unique name for each genre.

Members Table

member\_id: Primary Key, uniquely identifies each member.

email: Unique identifier for each member.

phone: Optional, unique identifier for each member.

Loans Table

loan\_id: Primary Key, uniquely identifies each loan.

book\_id: Foreign Key references Books.book\_id.

member\_id: Foreign Key references Members.member\_id.

due\_date: Must be later than the loan\_date.

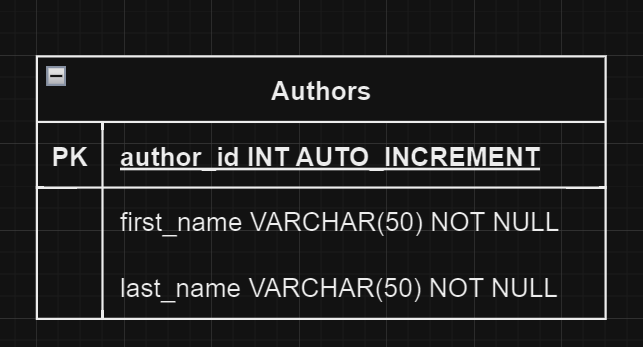
**SQL Statements to Create These tables are given Below:**  
  
CREATE TABLE Authors (

author\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL

);

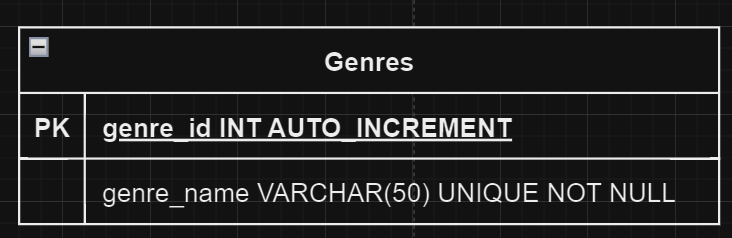


CREATE TABLE Genres (

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR(50) UNIQUE NOT NULL

);



CREATE TABLE Books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author\_id INT NOT NULL,

genre\_id INT NOT NULL,

isbn VARCHAR(13) UNIQUE NOT NULL,

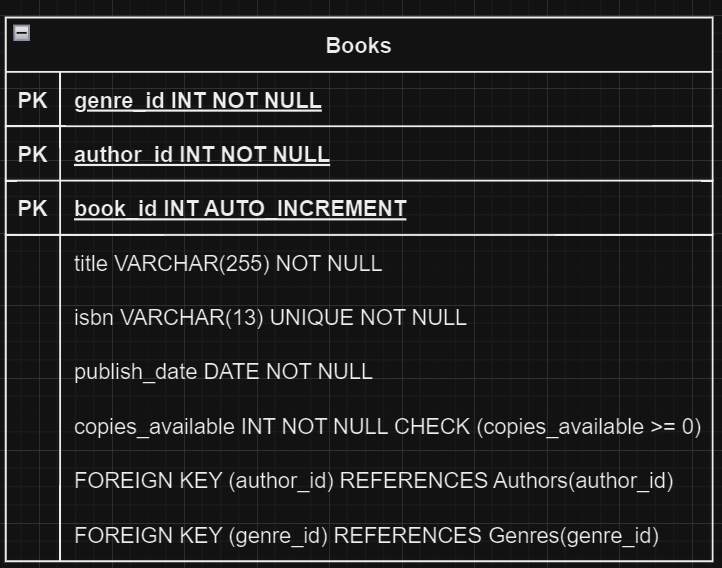
publish\_date DATE NOT NULL,

copies\_available INT NOT NULL CHECK (copies\_available >= 0),

FOREIGN KEY (author\_id) REFERENCES Authors(author\_id),

FOREIGN KEY (genre\_id) REFERENCES Genres(genre\_id)

);



CREATE TABLE Members (

member\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

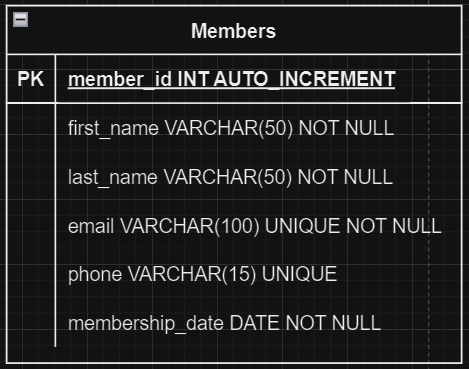
last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone VARCHAR(15) UNIQUE,

membership\_date DATE NOT NULL

);



CREATE TABLE Loans (

loan\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT NOT NULL,

member\_id INT NOT NULL,

loan\_date DATE NOT NULL,

return\_date DATE,

due\_date DATE NOT NULL CHECK (due\_date > loan\_date),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

FOREIGN KEY (member\_id) REFERENCES Members(member\_id)

);

