Library Booking System: Database Schema Documentation

Database CS A

Made by:

Klara Ahinta Daniswara	**24/532751/PA/22531**
Daffa Aryza Pasya	**24/532884/PA/22549**
Shadiq Arif Aryasatya	**24/536702/PA/22763**

DEPARTEMEN ILMU KOMPUTER dan ELEKTRONIKA

FAKULTAS MATEMATIKA dan ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2025

Tables of Content

Introduction

A. Purpose

This document explains the database structure for the Library Booking System. It covers each table, its attributes, and how the tables connect. The goal is to give a clear view of how data is stored and linked.

B. Overview of the Database Structure

The database has six main tables Book, BookCopy, Member, Staff, BorrowDetail, and Fine. Each table has a specific role. Foreign keys link the tables to support borrowing and fine processing. The structure follows 3NF to avoid repeated data and keep updates simple.

Entity and Table Descriptions

1.1. Book

Stores information about a book title.
Attributes include ISBN, title, genre, author names, publisher, and publication year.
Each ISBN is unique. A book can have several physical copies in the BookCopy table.

1.2. Member

Holds member data such as name, date of birth, contact details, join date, expiry date, and account status.
Email is unique. Each member can borrow many book copies through BorrowDetail.

1.3. Staff

Stores staff accounts.
Includes staff name, email, role, phone number, employment status, and password.
Email is unique. Staff handle book copies, borrowing processes, and fines.

1.4. BookCopy

Represents a physical copy of a book.
Each copy has a BookID and an availability status.
Linked to a book by ISBN.
Linked to the staff member responsible for the copy.

1.5. BorrowDetail

Tracks each borrowing action.
Includes borrow date, due date, return date, and references to BookCopy, Member, and Staff.
One record represents one borrow event.

1.6. Fine

Records fines issued for late returns.
Includes fine amount, issue date, status, and optional payment date.
Linked to a borrow record and the staff who handled the fine.

Relationships Overview

Key Relationships

- A book has many copies.
- A member can have many borrow transactions.
- A staff member can manage many borrow transactions.
- A borrow record can have one fine.
- A staff member can issue many fines.

Cardinality Summary

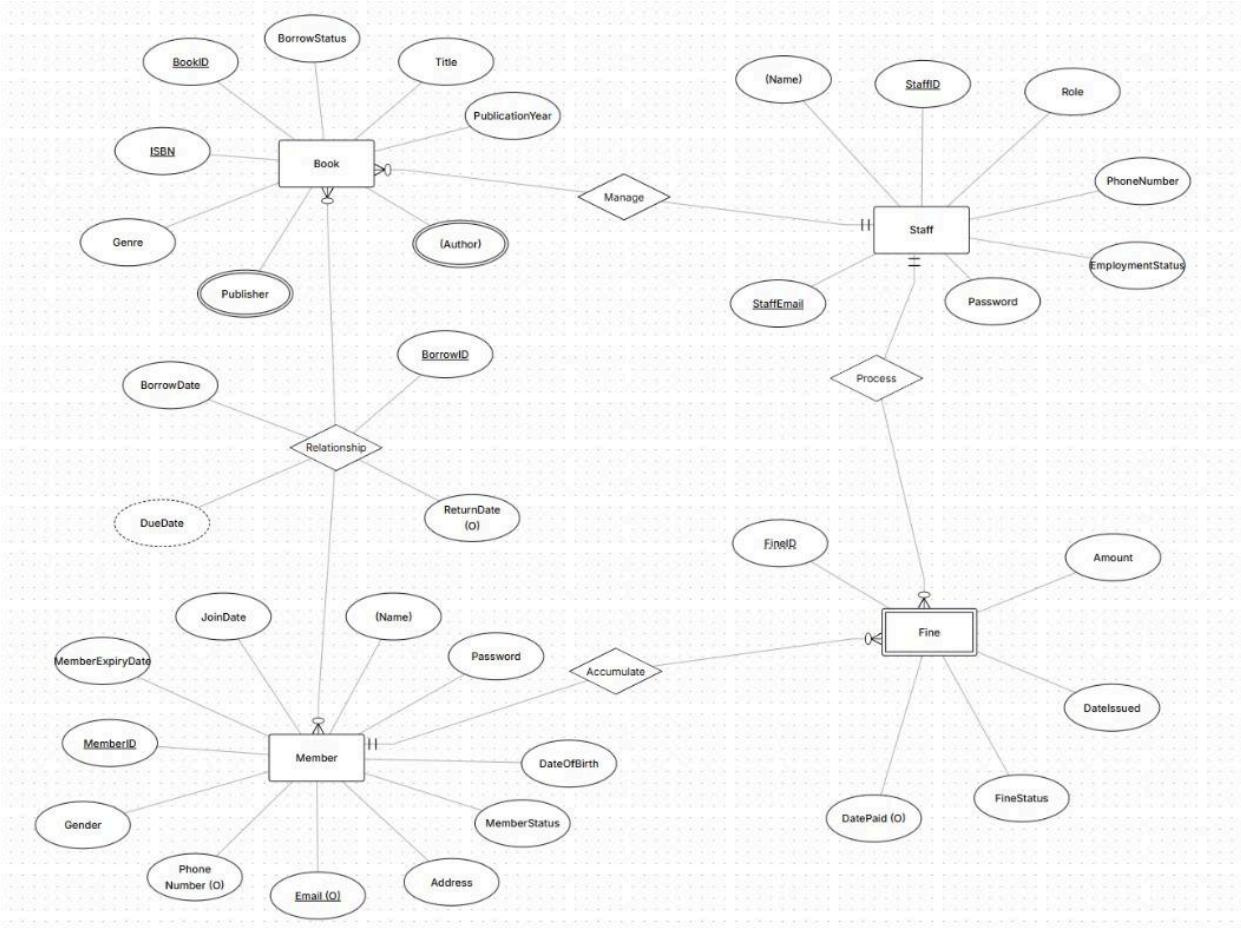
- Book to BookCopy: 1 to many
- Member to BorrowDetail: 1 to many
- Staff to BorrowDetail: 1 to many
- BorrowDetail to Fine: 1 to 1
- Staff to Fine: 1 to many

Relational Schema Diagram

This section shows the full relational schema used in the system.
It includes primary keys, foreign keys, and field types as shown in:

ERD Diagram

This section shows the updated ERD that maps all entities and their links.
It helps explain the flow between books, members, staff, borrow records, and fines.



Appendix

SQL Table Definitions

This section lists the SQL used to create all tables.
It includes field types, keys, and constraints.

```
CREATE TABLE Book
(
  Genre VARCHAR(255) NOT NULL,
  ISBN INT NOT NULL,
  Title VARCHAR(255) NOT NULL,
  PublicationYear VARCHAR(255) NOT NULL,
  Publisher VARCHAR(255) NOT NULL,
  AuthorFirstName VARCHAR(255) NOT NULL,
  AuthorLastName VARCHAR(255) NOT NULL,
  PRIMARY KEY (ISBN),
  UNIQUE (ISBN)
);

CREATE TABLE Member
(
  MemberID INT NOT NULL,
  MemberExpiryDate DATE NOT NULL,
  JoinDate DATE NOT NULL,
  Password VARCHAR(255) NOT NULL,
  DateOfBirth DATE NOT NULL,
  MemberStatus ENUM('Active', 'Not Active') NOT NULL,
  Address VARCHAR(255) NOT NULL,
  Email VARCHAR(255),
  PhoneNumber VARCHAR(30),
  Gender ENUM('Male', 'Female') NOT NULL,
  MemberFirstName VARCHAR(255) NOT NULL,
  MemberLastName VARCHAR(255) NOT NULL,
  PRIMARY KEY (MemberID),
  UNIQUE (Email)
);

CREATE TABLE Staff
(
  StaffID INT NOT NULL,
  StaffEmail VARCHAR(255) NOT NULL,
  Role VARCHAR(255) NOT NULL,
  Password VARCHAR(255) NOT NULL,
  EmploymentStatus ENUM('Active', 'Not Active') NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
```

```
  StaffFirstName VARCHAR(255) NOT NULL,
  StaffLastName VARCHAR(255) NOT NULL,
  PRIMARY KEY (StaffID),
  UNIQUE (StaffEmail)
);

CREATE TABLE BookCopy
(
  BookID INT NOT NULL,
  Availability ENUM('Available', 'Borrowed', 'Unavailable') NOT NULL,
  ISBN INT NOT NULL,
  StaffID INT NOT NULL,
  PRIMARY KEY (BookID),
  FOREIGN KEY (ISBN) REFERENCES Book(ISBN),
  FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);

CREATE TABLE BorrowDetail
(
  BorrowID INT NOT NULL,
  BorrowDate DATE NOT NULL,
  DueDate DATE NOT NULL,
  ReturnDate DATE,
  BookID INT NOT NULL,
  StaffID INT NOT NULL,
  MemberID INT NOT NULL,
  PRIMARY KEY (BorrowID),
  FOREIGN KEY (BookID) REFERENCES BookCopy(BookID),
  FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
  FOREIGN KEY (MemberID) REFERENCES Member(MemberID)
);

CREATE TABLE Fine
(
  FineID INT NOT NULL,
  Amount INT NOT NULL,
  DateIssued DATE NOT NULL,
  FineStatus ENUM('Paid', 'Not Paid') NOT NULL,
  DatePaid DATE,
  StaffID INT NOT NULL,
  BorrowID INT NOT NULL,
  PRIMARY KEY (FineID),
  FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
  FOREIGN KEY (BorrowID) REFERENCES BorrowDetail(BorrowID)
);
```

Notes and Constraints

- All IDs use primary keys.
- Email fields in Member and Staff are unique.
- Foreign keys enforce valid references between tables.
- Null fields are used only when needed, such as ReturnDate and DatePaid.