# Simple Library Management System

## 1. Project Overview

This project is a beginner-level SQL exercise focused on building a basic Library Management System using SQL. It involves creating a relational database with tables to store information about books, borrowers, and borrowed records.

## 2. Database Structure

Database Name: LibraryDB

* Tables:
* Books: Stores information about books available in the library.
* Columns: book\_id (Primary Key), title, author, genre, published\_year
* Borrowers: Contains details about library members who borrow books.
* Columns: borrower\_id (Primary Key), name, email, phone\_number
* BorrowedBooks: Tracks book borrowing transactions and return dates.
* Columns: borrowed\_id (Primary Key), book\_id (Foreign Key referencing Books.book\_id), borrower\_id (Foreign Key referencing Borrowers.borrower\_id), borrow\_date, return\_date

## 3. Relationships Between Tables

Books to BorrowedBooks:  
One-to-Many relationship. Each book can appear in the BorrowedBooks table multiple times if borrowed by different borrowers or on different dates.

Borrowers to BorrowedBooks:  
One-to-Many relationship. Each borrower can have multiple entries in the BorrowedBooks table for each book they borrow.

These relationships are maintained using Foreign Key constraints to ensure that each borrowed record links to an existing book and borrower.

## 4. SQL Queries Performed

* Inserting Data:
* Insert sample data into Books, Borrowers, and BorrowedBooks to set up initial records.
* Select Queries:
* List all books:

SELECT \* FROM Books;

* List all borrowers:

SELECT \* FROM Borrowers;

* List all borrowed books with details:

SELECT b.title, br.name, bb.borrow\_date, bb.return\_date  
FROM BorrowedBooks bb  
JOIN Books b ON bb.book\_id = b.book\_id  
JOIN Borrowers br ON bb.borrower\_id = br.borrower\_id;

* Update Operation:
* Update return\_date when a book is returned:

UPDATE BorrowedBooks  
SET return\_date = 'YYYY-MM-DD'  
WHERE borrowed\_id = X;

* Delete Operation:
* Remove a borrower record if they have no outstanding borrowed books:

DELETE FROM Borrowers  
WHERE borrower\_id = X  
AND NOT EXISTS (  
 SELECT 1 FROM BorrowedBooks WHERE borrower\_id = X AND return\_date IS NULL  
);

## 5. Learning Outcomes

Creating and Relating Tables: Built an understanding of primary and foreign key constraints.

Basic CRUD Operations: Practiced inserting, selecting, updating, and deleting records.

Using Foreign Keys: Learned how to manage relationships between tables to ensure data integrity.