Library Management System - SQL Project

A simple MySQL-based Library Management System for managing books, users, issued books, returns, and fines.

Features:

- Add books and users (students or staff)

- Issue and return books

- Track book availability

- Calculate overdue fines

- View overdue books

Technologies Used:

- MySQL (SQL only)

- (Optional: Python/Java/HTML for interface)

Database Structure:

```
CREATE TABLE books (
    book_id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100),
    author VARCHAR(100),
    genre VARCHAR(50),
    total_stock INT,
    available_stock INT
);
```

```
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    email VARCHAR(100),
    user_type ENUM('student', 'staff')
);

CREATE TABLE issue_book (
    issue_id INT PRIMARY KEY AUTO_INCREMENT,
    book_id INT,
    user_id INT,
    issue_date DATE,
    due_date DATE,
    FOREIGN KEY (book_id) REFERENCES books(book_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE return_book (
    return_id INT PRIMARY KEY AUTO_INCREMENT,
    issue_id INT,
    return_date DATE,
    fine DECIMAL(5,2),
    FOREIGN KEY (issue_id) REFERENCES issue_book(issue_id)
);
```

Sample Queries:

```
-- Issue Book
INSERT INTO issue_book (book_id, user_id, issue_date, due_date)
VALUES (1, 1, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 14 DAY));

UPDATE books SET available_stock = available_stock - 1 WHERE book_id = 1;

-- Return Book
INSERT INTO return_book (issue_id, return_date, fine)
VALUES (1, CURDATE(),
  CASE
    WHEN CURDATE() > (SELECT due_date FROM issue_book WHERE issue_id = 1)
      THEN DATEDIFF(CURDATE(), (SELECT due_date FROM issue_book WHERE issue_id = 1)) * 2
    ELSE 0
  END);

UPDATE books SET available_stock = available_stock + 1
WHERE book_id = (SELECT book_id FROM issue_book WHERE issue_id = 1);

-- View Available Books
SELECT * FROM books WHERE available_stock > 0;
```

```
-- Overdue Books
SELECT i.issue_id, u.name, b.title, i.due_date
FROM issue_book i
JOIN users u ON i.user_id = u.user_id
JOIN books b ON i.book_id = b.book_id
WHERE i.issue_id NOT IN (SELECT issue_id FROM return_book)
AND i.due_date < CURDATE();
```

Author:

[Your Name]