Ý tưởng

**Yêu cầu chức năng:**

Quản lý Sách:

Hệ thống cần cho phép thêm, sửa, xóa và xem thông tin về các cuốn sách trong thư viện.

Mỗi cuốn sách có các thuộc tính như: Mã sách, Tên sách, Tác giả, Thể loại, Số lượng hiện có.

Quản lý Thẻ Thư viện:

Hệ thống cần cho phép quản lý thông tin về các thẻ thư viện của người đọc.

Mỗi thẻ thư viện có các thuộc tính như: Mã thẻ, Họ và tên người đọc, Địa chỉ, Số điện thoại.

Quản lý Mượn trả:

Người đọc có thể mượn sách từ thư viện và trả lại sau một khoảng thời gian.

Mỗi giao dịch mượn trả bao gồm thông tin về sách được mượn, người mượn, ngày mượn và ngày hết hạn.

**Mô hình cơ sở dữ liệu:**

Bảng "Books":

book\_id: (Primary Key) – varchar(5) - Mã sách.

title: NVarchar(255) - Tên sách.

author: NVarchar(100) - Tác giả của sách.

category: NVarchar(50) - Thể loại của sách.

quantity\_available: Integer - Số lượng sách hiện có.

Mô tả: Text

Giá: Decimal(10, 2)

Năm xuất bản: int

Bảng "Library\_Cards":

card\_id: (Primary Key) – varchar(5) - Mã thẻ thư viện.

full\_name: NVarchar(100) - Họ và tên chủ thẻ.

address: NVarchar(255) - Địa chỉ của chủ thẻ.

phone\_number: Varchar(10) - Số điện thoại của chủ thẻ.

Email: varchar(255)

Ngày đăng ký: Date

Ngày hết hạn: Date

Bảng "Loans":

loan\_id: (Primary Key) – varchar(5) - Mã giao dịch mượn trả.

book\_id: (Foreign Key) – varchar(5) - Mã sách (liên kết với bảng Books).

card\_id: (Foreign Key) – varchar(5) - Mã thẻ thư viện (liên kết với bảng Library\_Cards).

loan\_date: Date - Ngày mượn sách.

due\_date: Date - Ngày trả sách dự kiến.

số tiền phạt: Decimal(10,2)

trạng thái: Nvarchar(50)

**Ghi chú:**

Mỗi giao dịch mượn trả được ghi lại trong bảng "Loans", với mỗi bản ghi thể hiện một cuốn sách được mượn bởi một người đọc cụ thể.

Mỗi cuốn sách có thể được mượn bởi nhiều người khác nhau, do đó mối quan hệ giữa Sách và Thẻ Thư viện là mối quan hệ 1- N.

----Phần này làm trong lúc làm hệ thống--- (để hthanh làm)  
  
**Ràng buộc số lượng sách không âm:** Trong bảng "Books", bạn muốn đảm bảo rằng số lượng sách không được âm.

CREATE TRIGGER check\_quantity\_non\_negative

BEFORE UPDATE OF quantity\_available ON Books

FOR EACH ROW

BEGIN

IF NEW.quantity\_available < 0 THEN

RAISE (ABORT, 'Số lượng sách không được âm');

END IF;

END;

**Ràng buộc ngày mượn và ngày trả**: Trong bảng "Loans", bạn muốn đảm bảo rằng ngày mượn phải trước ngày hết hạn.

CREATE TRIGGER check\_loan\_dates

BEFORE INSERT ON Loans

FOR EACH ROW

BEGIN

IF NEW.loan\_date > NEW.due\_date THEN

RAISE (ABORT, 'Ngày mượn phải trước ngày hết hạn');

END IF;

END;

**Ràng buộc mượn sách chỉ khi sách còn tồn tại:** Trong bảng "Loans", bạn muốn đảm bảo rằng sách chỉ có thể được mượn khi còn tồn tại ít nhất một cuốn sách trong thư viện.

CREATE TRIGGER check\_book\_availability

BEFORE INSERT ON Loans

FOR EACH ROW

BEGIN

DECLARE available\_quantity INTEGER;

SELECT quantity\_available INTO available\_quantity FROM Books WHERE book\_id = NEW.book\_id;

IF available\_quantity IS NULL OR available\_quantity <= 0 THEN

RAISE (ABORT, 'Sách không còn tồn tại trong thư viện');

END IF;

END;

**Ràng buộc xóa thẻ thư viện khi không có mượn trả nào liên kết**: Trong bảng "Library\_Cards", bạn muốn đảm bảo rằng một thẻ thư viện chỉ có thể bị xóa khi không còn bất kỳ giao dịch mượn trả nào liên kết với nó.

CREATE TRIGGER prevent\_delete\_library\_card

BEFORE DELETE ON Library\_Cards

FOR EACH ROW

BEGIN

DECLARE loan\_count INTEGER;

SELECT COUNT(\*) INTO loan\_count FROM Loans WHERE card\_id = OLD.card\_id;

IF loan\_count > 0 THEN

RAISE (ABORT, 'Không thể xóa thẻ thư viện có giao dịch mượn trả liên kết');

END IF;

END;