**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN AN TOÀN THÔNG TIN**

**🙚🙤🕮🙦🙘**

****

**BÁO CÁO ĐỀ TÀI**

**QUẢN LÝ THƯ VIỆN**

**MÃ MÔN HỌC: DBMS330284\_23\_1\_03**

**THỰC HIỆN: Nhóm 8**

**Thành viên:**

21110368 - Nguyễn Thị Lan Anh

21110562 - Hoàng Võ Ngọc Nguyên

21110645 - Nguyễn Hồ Thiên Thanh

21110871 - Nguyễn Nhật Duy

**GIẢNG VIÊN HƯỚNG DẪN: TS. Nguyễn Thành Sơn**

**Tp. Hồ Chí Minh, tháng 11 năm 20****23**

MỤC LỤC

[**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG** 4](#_Toc151678434)

[**1.** **Mô tả bài toán** 4](#_Toc151678435)

[**1.1.** **Tổng quan:** 4](#_Toc151678436)

[**1.2.** **Đặc điểm:** 4](#_Toc151678437)

[**1.3.** **Lợi ích:** 5](#_Toc151678438)

[**2.** **Nghiệp vụ bài toán** 6](#_Toc151678439)

[**3.** **Chức năng** 6](#_Toc151678440)

[**4.** **Tính tiền vi phạm nội quy mượn trả sách** 7](#_Toc151678441)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 11](#_Toc151678442)

[**1.** **Thiết kế cơ sở dữ liệu mức logic** 11](#_Toc151678443)

[**2.** **Các ràng buộc cần có** 12](#_Toc151678444)

[**3.** **Cài đặt CSDL và các ràng buộc đơn giản** 13](#_Toc151678445)

[**4.** **Trigger** 15](#_Toc151678446)

[**5.** **Views** 17](#_Toc151678447)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG** 19](#_Toc151678448)

[**1.** **Kết nối cơ sở liệu** 19](#_Toc151678449)

[**2.** **Xem thông tin danh mục** 19](#_Toc151678450)

[**3.** **Quản lý sách** 21](#_Toc151678451)

[**3.1.** **Tìm kiếm chi tiết sách** 21](#_Toc151678452)

[**3.2.** **Tạo sách mới thêm vào bảng chi tiết sách** 24](#_Toc151678453)

[**3.3.** **Chỉnh sửa thông tin sách** 25](#_Toc151678454)

[**3.4.** **Xóa sách** 27](#_Toc151678455)

[**4.** **Quản lí độc giả** 27](#_Toc151678456)

[**4.1.** **Thêm độc giả mới** 27](#_Toc151678457)

[**4.2.** **Sửa thông tin độc giả** 29](#_Toc151678458)

[**4.3.** **Xóa độc giả** 30](#_Toc151678459)

[**4.4.** **Tìm kiếm độc giả** 31](#_Toc151678460)

[**5.** **Quản lý phiếu mượn trả** 33](#_Toc151678461)

[**5.1.** **Thêm phiếu mượn** 33](#_Toc151678462)

[**5.2.** **Cập nhật phiếu mượn** 35](#_Toc151678463)

[**5.3.** **Xem phiếu mượn** 36](#_Toc151678464)

[**5.4.** **Tìm kiếm phiếu mượn** 36](#_Toc151678465)

[**6.** **Quản lý phiếu trả (Chi tiết phiếu mượn trả)** 39](#_Toc151678466)

[**6.1.** **Thêm phiếu trả** 40](#_Toc151678467)

[**6.2.** **Cập nhật Phiếu Trả** 41](#_Toc151678468)

[**6.3.** **Xem phiếu Trả** 42](#_Toc151678469)

[**6.4.** **Tìm kiếm phiếu trả** 43](#_Toc151678470)

[**6.5.** **Xóa phiếu trả** 44](#_Toc151678471)

[**7.** **Quản lý nhân viên** 45](#_Toc151678472)

[**7.1.** **Thêm nhân viên** 45](#_Toc151678473)

[**7.2.** **Cập nhật nhân viên** 47](#_Toc151678474)

[**7.3.** **Xem nhân viên** 48](#_Toc151678475)

[**7.4.** **Tìm kiếm nhân viên** 49](#_Toc151678476)

[**7.5.** **Xóa Nhân viên** 51](#_Toc151678477)

[**8.** **Quản lý thống kê** 52](#_Toc151678478)

[**8.1.** **Thống kê sách** 52](#_Toc151678479)

[**8.2.** **Thống kê độc giả** 54](#_Toc151678480)

[**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN** 57](#_Toc151678481)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG** 60](#_Toc151678482)

[**1.** **Trang login** 60](#_Toc151678483)

[**2.** **Trang home** 62](#_Toc151678484)

[**3.** **Trang quản lý sách** 62](#_Toc151678485)

[**4.** **Trang quản lý độc giả** 63](#_Toc151678486)

[**5.** **Trang quản lý phiếu mượn trả** 64](#_Toc151678487)

[**5.1.** **Quản lý phiếu mượn** 64](#_Toc151678488)

[**5.2.** **Quản lý phiếu trả (Chi tiết phiếu mượn trả)** 65](#_Toc151678489)

[**6.** **Quản lý nhân viên** 65](#_Toc151678490)

[**7.** **Trang quản lý thống kê** 66](#_Toc151678491)

[**7.1.** **Thống kê sách** 66](#_Toc151678492)

[**7.1.1.** **Thống kê sách được mượn nhiều nhất** 66](#_Toc151678493)

[**7.1.2.** **Thống kê sách theo thể loại** 67](#_Toc151678494)

[**7.1.3.** **Thống kê sách theo tác giả** 69](#_Toc151678495)

[**7.2.** **Thống kê độc giả** 70](#_Toc151678496)

[**7.2.1.** **Thống kê độc giả trả trễ hạn** 70](#_Toc151678497)

[**7.2.2.** **Thống kê độc giả làm hư sách** 71](#_Toc151678498)

[**7.2.3.** **Thống kê độc giả làm mất sách** 72](#_Toc151678499)

[**8.** **Trang tài khoản (Cho phép đổi password)** 73](#_Toc151678500)

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

## **Mô tả bài toán**

### **Tổng quan:**

"Quản Lý Thư Viện" là một dự án hệ thống quản lý được phát triển nhằm hỗ trợ quá trình quản lý và vận hành trong một thư viện. Dự án này được xây dựng để giúp thư viện theo dõi tài liệu và quản lý độc giả một cách hiệu quả và tiện lợi. Mục tiêu chính của dự án "Quản Lý Thư Viện" là tạo ra một hệ thống quản lý cho các hoạt động mượn trả sách trong thư viện. Hệ thống này sẽ giúp đơn giản hóa các nhiệm vụ quản lý hàng ngày, giúp người quản lý thư viện tiết kiệm thời gian đồng thời nâng cao chất lượng dịch vụ đối với độc giả.

### **Đặc điểm:**

Chương trình "Quản Lý Thư Viện" được xây dựng với mục tiêu phục vụ đối tượng người dùng gồm người quản lý thư viện và các nhân viên trực thư viện, giúp họ quản lý và theo dõi các hoạt động trong thư viện một cách hiệu quả và thuận tiện. Đối với hệ thống quản lý này cần lưu trữ những dữ liệu sau:

**Đầu sách:** Lưu trữ các thông tin bao gồm: Mã sách, tên sách, mã thể loại, mã NXB, giá, số lượng. Mã sách dùng để định danh duy nhất cho mỗi Đầu sách. Mỗi đầu sách bao gồm bởi một hoặc nhiều cuốn sách, và được sáng tác bới một hoặc nhiều tác giả, và thuộc trong một hoặc nhiều thể loại và được xuất bản bởi duy nhất một nhà xuất bản.

**Cuốn sách**: Lưu trữ các thông tin bao gồm: Mã cuốn, mã sách. Mã cuốn dùng để định danh duy nhất cho một cuốn sách. Mỗi cuốn sách chỉ thuộc duy nhất một đầu sách và chỉ được đại diện duy nhất bởi một phiếu mượn trả.

**Độc giả:** Lưu trữ các thông tin bao gồm: Mã độc giả, tên độc giả, giới tính, số CCCD, số điện thoại, địa chỉ, email. Mã độc giả dùng để định danh duy nhất cho một độc giả. Mỗi độc giả được cung cấp một hoặc nhiều phiếu mượn trả.

**Phiếu** **Mượn trả**: Lưu trữ các thông tin bao gồm: Mã phiếu, mã nhân viên, mã độc giả, ngày mượn, ngày hết hạn. Mã phiếu được sử dụng để định danh duy nhất cho mỗi phiếu mượn trả, và mỗi mã phiếu chỉ được cung cấp duy nhất cho một độc giả, mỗi phiếu mượn trả có thể đại diện cho một hoặc nhiều cuốn sách. Mỗi phiếu mượn 1trả được ghi nhận bởi duy nhất một nhân viên.

**CT mượn trả:** Lưu trữ các thông tin bao gồm: Mã cuốn, mã phiếu, ngày trả, tình trạng, số ngày quá hạn, ghi chú, tiền phạt, thanh toán. Mã cuốn được sử dụng để định danh duy nhất cho mỗi CT mượn trả. Tình trạng sách được định nghĩa là trạng thái của sách sau khi độc giả tới thư viện và trả sách, tình trạng sách được biểu diễn bằng 3 giá trị bao gồm: "Hư, Mất, BT", với Hư là trạng thái sách bị hư hỏng, Mất là sách bị mất, BT là sách bình thường. Ghi chú bao gồm các thông tin chi tiết về tình trạng sách, có thể chứa thông tin ghi chú cần thiết khác liên quan đến trạng thái sau khi trả sách.

**Nhân viên**: Lưu trữ các thông tin bao gồm: Mã nhân viên, họ tên, CCCD, Số điện thoại, Địa chỉ, Email. Mã nhân viên dùng để định danh duy nhất cho mỗi nhân viên và mỗi nhân viên ghi nhận cho một hoặc nhiều phiếu mượn trả.

Thư viện còn có phần thống kê sách, cung cấp các thông tin tổng hợp sách, độc giả. Thống kê sách được mượn nhiều nhất, sách theo thể loại, theo tác giả. Thống kê độc giả vi phạm theo tình trạng như: “ Trễ hạn, hư sách, mất sách”.

* ***Nội Quy Vi Phạm Mượn Trả Sách:***

Tiền vi phạm được tính toán dựa trên số lỗi vi phạm có trong phiếu vi phạm được cung cấp cho độc giả.

- *Trong đó:*

+ Hạn trả cho tất cả các thể loại sách là 30 ngày kể từ ngày độc giả mượn sách hợp lệ trên phiếu mượn.

+ Khi trả sách trễ so với ngày hẹn trả thì độc giả phải chịu mức phạt 1000đ/ngày cho mỗi sách.

+ Nếu làm hư hỏng sách dưới 5 trang sẽ phải chịu phạt số tiền bằng 50% giá trị của sách mượn.

+ Nếu làm hư hỏng sách trên 5 trang bao gồm cả việc rách sách, vẽ vào sách, bẩn sách,... sẽ quy vào tình trạng mất sách và phải đền bù theo nội quy đã định.

+ Nếu làm mất sách, phải đền đúng sách đã mượn.

+ Nếu làm mất nhưng không thể đền đúng sách thì phải bồi thường gấp 3 lần giá trị sách mượn gốc.

+ Nếu quá hạn 30 ngày kể từ hạn trả sách mà độc giả chưa trả sách, thư viện có thể liên hệ với cơ quan pháp luật hoặc cơ quan an ninh để xác định vị trí của độc giả và sử dụng các biện pháp pháp lý để đòi lại sách và tiền bồi thường từ độc giả bỏ trốn.

+ Trường hợp độc giả đang vi phạm nội quy sẽ không được mượn sách từ thư viện cho đến khi thư viện xác nhận độc giả đã hoàn thành việc bồi thường.

+ Thư viện sẽ đưa ra số tiền phải bồi thường dựa vào nội quy vi phạm của thư viện. Số tiền vi phạm của độc giả bằng tổng số tiền đã vi phạm theo nội quy vi phạm mượn trả sách của thư viện.

* 1. **Lợi ích:**

- *Tối ưu hóa hoạt động*: Giúp thư viện quản lý tài liệu, độc giả một cách hiệu quả hơn, tiết kiệm thời gian và tối ưu hóa quá trình làm việc.

- *Cải thiện dịch vụ*: Hệ thống giúp nâng cao chất lượng dịch vụ đối với độc giả thông qua việc cung cấp thông tin sách chính xác, thống kê tổng quan và quản lý mượn trả tốt hơn.

- *Minh bạch và đồng nhất*: Dự án đảm bảo tính minh bạch và đồng nhất trong việc.

## **Nghiệp vụ bài toán**

Bài toán quản lý thư viện là một nghiệp vụ quản lý thông tin trong lĩnh vực thư viện học trong ngành giáo dục. Lĩnh vực này liên quan đến nhiều việc quản lý độc giả, quản lý sách và điều hành việc tổ chức, cung cấp tài liệu bao gồm sách, tạp chí, và các nguồn thông tin khác đáp ứng nhu cầu sử dụng của độc giả. Để đảm bảo thư viện hoạt động một cách hiệu quả và cung cấp các dịch vụ tốt nhất yêu cầu nhân viên phải có trách nhiệm cực kỳ quan trọng đối với độc giả.

Để thực hiện một cách hiệu quả những nhiệm vụ này, việc đào tạo nghiệp vụ cho nhân viên thư viện là không thể thiếu. Đào tạo này cần tập trung vào việc nắm vững các quy trình và kỹ năng cần thiết trong quản lý sách, quản lý thông tin độc giả, và quy trình mượn trả sách và các nội quy định vi phạm mà thư viện đề ra. Nhân viên thư viện cũng cần đảm bảo việc liên tục cập nhật kiến thức về các xu hướng mới trong lĩnh vực thư viện học và công nghệ thông tin.

- ***Quy trình mượn sách****:* Khi đến thư viện mượn sách, độc giả có thể tìm sách trực tiếp trong thư viện hoặc có thể nói cho nhân viên thông tin sách muốn tìm để nhân viên có thể tìm giúp. Sau đó, độc giả sẽ mang sách đến bàn làm việc của nhân viên để làm thủ tục mượn sách. Nhân viên sẽ lưu lại các thông tin của độc giả và tiến hành in phiếu mượn, độc giả sẽ kiểm tra lại thông tin trên phiếu mượn trước khi ra khỏi thư viện. Độc giả phải trả lại sách đã mượn cho thư viện theo đúng thời hạn, nếu vi phạm sẽ phải chịu bồi thường theo nội quy đã định của thư viện.

- ***Quy trình trả sách:*** Độc giả mang theo sách họ đã mượn và quay lại thư viện để mượn trả sách. Độc giả cung cấp phiếu mượn trả tại quầy mượn trả. Nhân viên kiểm tra phiếu mượn trả và xác minh thông tin của độc giả. Nhân viên sẽ thực hiện việc kiểm tra tình trạng của sách, nếu tình trạng sách thuộc vi phạm trong nội quy thì nhân viên yêu cầu độc giả bồi thường dựa trên nội quy vi phạm của thư viện.

- ***Quy trình trả sách đóng tiền vi phạm:*** Đối với độc giả trả sách trễ, độc giả sẽ nhận được thông báo vi phạm từ thư viện thông qua email. Mức phạt có thể khác nhau tùy thuộc vào loại vi phạm và thời gian trễ trả sách. Sau khi nhận được thông tin, người mượn đến thư viện để đóng tiền vi phạm và trả sách, khi đó nhân viên sẽ kiểm tra tình hình sách (ví dụ: sách có hư hỏng hay mất không...) và tổng kết tính toán số tiền vi phạm của người mượn. Người mượn sách cần thanh toán số tiền phạt tại bộ phận quản lý của thư viện. Họ có thể sử dụng các phương thức thanh toán khác nhau như tiền mặt, thẻ tín dụng, hoặc chuyển khoản. Sau khi thanh toán, người mượn cần xác nhận với thư viện bằng cách cung cấp biên lai hoặc chứng từ thanh toán để chứng minh việc trả phạt đã được thực hiện. Khi số tiền phạt đã được thanh toán và xác nhận từ thư viện (đóng mộc vào phiếu thanh toán, kèm chữ ký của nhân viên chịu trách nhiệm trong thời gian đó), thư viện sẽ ghi nhận việc này. Độc giả sẽ được phép mượn sách tiếp theo phù hợp với quy định cho phép của nội quy thư viện.

## **Chức năng**

- **Quản lý sách:**

+ *Xem thông tin chi tiết*: Cho phép nhân viên xem các thông tin của sách (Mã sách, tên sách,Tác Giả, Nhà Xuất Bản, Thể loại, giá, số lượng).

+ *Thêm sách mới:* Thêm dữ liệu các thông tin về sách mới vào hệ thống.

+ *Xóa sách:* Xóa dữ liệu của các cuốn sách không còn sử dụng ra khỏi hệ thống.

+ Cập nhật, chỉnh sửa thông tin sách: Cho phép cập nhật thông tin sách đã có, sửa đổi các thông tin sách.

+ *Tìm kiếm:* Cho phép tìm kiếm sách theo các tiêu chí (tên sách , Tác giả, nhà xuất bản, thể loại).

- **Quản lý độc giả:**

+ *Thêm thông tin độc giả:* Cho phép nhân viên thêm thông tin của độc giả mới (mã độc giả, tên độc giả, giới tính, CMND, số điện thoại, địa chỉ, email).

+ *Tra cứu thông tin*: Cho phép nhân viên tra cứu của độc giả .

+ *Xóa:* Cho phép nhân viên xóa thông tin độc giả ra khỏi hệ thống.

+ *Cập nhật:* Cho phép nhân viên cập nhật lại thông tin của độc giả đã có.

+ *Lập phiếu mượn trả / vi phạm*: Cho phép nhân viên lập các phiếu mượn trả / vi phạm cho độc giả.

- **Phiếu mượn trả:**

+ *Ghi nhận thông tin về việc mượn trả sách* bao gồm thông tin độc giả, sách được mượn, ngày mượn, ngày trả, số lượng sách mà độc giả mượn.

+ *Tạo phiếu mượn trả sách* chứa thông tin chi tiết về giao dịch để dễ dàng kiểm tra và quản lý.

+ *Xem lịch sử mượn trả:* Hiển thị các lịch sử giao dịch mượn trả của các độc giả.

+ *Xử lý vi phạm:* Ghi nhận các tình trạng vi phạm như quá hạn, bị hỏng,... tính toán chi phí đền bù.

- **Phiếu vi phạm:**

+ Ghi nhận thông tin về việc hư hỏng/mất/trả quá hạn sách từ độc giả.

+ Lập phiếu vi phạm độc giả khi độc giả vi phạm quá mức quy định.

- **Thống kê:**

+ *Thống kê sách:* Thống kê số lượng theo các tiêu chí (tác giả, thể loại, )số sách được mượn nhiều nhất, số lượng sách còn lại.

+ *Thống kê độc giả:* Cho biết tình hình mượn trả của các độc giả vi phạm như: Trả trễ sách, Hỏng sách, Mất sách.

## **Tính tiền vi phạm nội quy mượn trả sách**

Để tính tiền vi phạm nội quy mượn sách của độc giả, cần các yếu tố sau:

- Phí bồi thường cho sách trả quá hạn bằng **1 nghìn đồng** cho mỗi cuốn sách trả quá hạn.

- Phí bồi thường cho sách bị hư hỏng bằng **50%** giá tiền của cho mỗi cuốn sách bị hư hỏng.

- Phí bồi thường cho sách bị mất sẽ bằng **300%** giá tiền cho mỗi cuốn sách bị mất.

→ Dựa vào các thông tin trên, tổng tiền vi phạm của độc giả (S) có thể được tính bằng công thức sau:

**S = 1\*S1 + 0.5\*S2 + 3\*S3**

*(đơn vị: Nghìn đồng)*

● ***Trong đó:***

- **S1** là tổng số ngày quá hạn: Đây là tổng số ngày mà độc giả trễ việc trả lại sách cho thư viện so với ngày trả đã cam kết ở phiếu mượn ban đầu.

- **S2** là tổng tiền sách bị hỏng : Đây là tổng số tiền sách vi phạm do độc giả đã gây hư hỏng khi mượn theo quy định của nội quy Thư viện.

- **S3** là tổng số tiền sách bị mất: Đây là tổng số tiền sách vi phạm do độc giả đã làm mất sách khi mượn theo quy định của nội quy Thư viện.

***Lưu Ý:***

- *Đối với độc giả lựa chọn mua lại sách cho thư viện khi làm mất sách, giá tiền phạt của cuốn sách làm mất đó sẽ bằng* ***0 đồng*** *cho cả trường hợp còn hạn và quá hạn. Đối với độc giả trả sách bị hư hỏng* ***hơn 5 trang*** *so với chất lượng sách ban đầu sẽ quy về tình trạng mất sách và phải tính tiền phạt hoặc đền bù theo nội quy.*

**\* Ví dụ:** Giả sử có 2 độc giả A, B mượn sách. Sau khi quá hạn trả sách ở Thư viện thì 2 độc giả A, B trả lại sách với tình trạng:

* **Độc giả A:** 1 cuốn sách có giá 170 nghìn đồng, bị rách 2 trang, quá hạn 9 ngày kể từ ngày phải trả.
* **Độc giả B:** 2 cuốn sách, trong đó có 1 cuốn bị mất có giá 200 nghìn đồng (có mua sách giống bản gốc đền lại cho thư viện), 1 cuốn bị hư hỏng 1 trang có giá 150 nghìn đồng, cả 2 cuốn đều quá hạn 7 ngày.

***Tính số tiền vi phạm của độc giả A và B.***

***Bài giải:***

Ta có:

- **Độc giả A:**

+ Giá sách: 170 nghìn đồng.

+ Tổng số ngày quá hạn S1 = 9 ngày

+ Tổng số tiền sách bị hỏng S2 = 170 nghìn đồng

+ Tổng số tiền sách bị mất S3 = 0 nghìn đồng

→ Số tiền vi phạm của độc giả A:

**S = 1\*S1 + 0.5\*S2 + 3\*S3** = 1\*9 + 0.5\*170 + 3\*0 = 94 nghìn đồng

- **Độc giả B:**

+ Giá sách: Cuốn 1 - 200 nghìn đồng.

Cuốn 2 - 150 nghìn đồng.

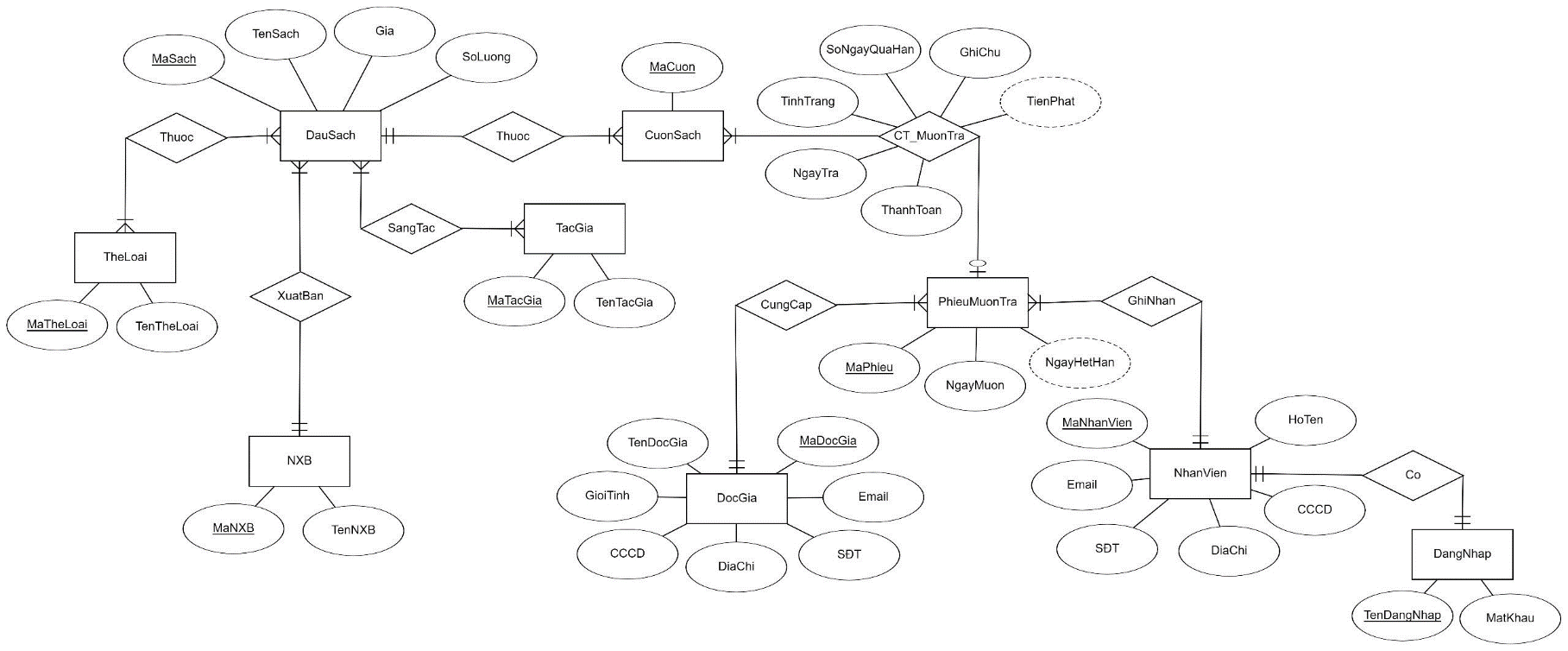
+ Tổng số ngày quá hạn S1 = 7 + 7 = 14 ngày

+ Tổng số tiền sách bị hỏng S2 = 150 nghìn đồng

+ Tổng số tiền sách bị mất S3 = 0 nghìn đồng *( đã mua lại sách mới)*

→ Số tiền vi phạm của độc giả B:

**S = 1\*S1 + 0.5\*S2 + 3\*S3** = 1\*14 + 0.5\*150 + 3\*0 = 89 nghìn đồng



# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **Thiết kế cơ sở dữ liệu mức logic**

* DauSach (MaSach, TenSach, MaTacGia, MaTheLoai, MaNXB, Gia, SoLuong)
* CuonSach (MaCuon, MaSach)
* TacGia (MaTacGia, TenTacGia)
* SangTac (MaSach, MaTacGia)
* TheLoai (MaTheLoai, TenTheLoai)
* DauSach\_TheLoai (MaSach, MaTheLoai)
* NXB (MaNXB, TenNXB)
* DocGia(MaDocGia, TenDocGia, GioiTinh, CCCD, DiaChi, SDT, Email)
* PhieuMuonTra(MaPhieu, MaDocGia, MaNhanVien, NgayMuon, NgayHetHan)
* CT\_MuonTra(MaCuon, MaPhieu, NgayTra, TinhTrang, SoNgayQuaHan, GhiChu, TienPhat, ThanhToan)
* NhanVien (MaNhanVien, HoTen, CCCD, SDT, DiaChi, Email)
* DangNhap(TenDangNhap,MatKhau)

**Diagram**



## **Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | DauSach | - Khóa chính MaSach  - Khóa ngoại MaTacGia tham chiếu đến bảng TacGia  - Khóa ngoại MaTheLoai tham chiếu đến bảng TheLoai  - Khóa ngoại MaNXB tham chiếu đến bảng NXB  - CHECK Gia >= 0  - CHECK SoLuong >= 0 |
| 2 | CuonSach | - Khóa chính MaCuon  - Khóa ngoại MaSach tham chiếu đến bảng DauSach |
| 3 | TacGia | - Khóa chính MaTacGia  - NOT NULL TenTacGia |
| 4 | SangTac | - Khóa chính MaSach, MaTacGia |
| 5 | TheLoai | - Khóa chính MaTheLoai  - NOT NULL TenTheLoai |
| 6 | DauSach\_TheLoai | - Khóa chính MaSach, MaTheLoai |
| 5 | NXB | - Khóa chính MaNXB  - NOT NULL TenNXB |
| 6 | ĐocGia | - Khóa chính MaDocGia  - NOT NULL TenDocGia  - CHECK GioiTinh IN (‘Nam’, ‘Nữ’)  - CHECK (LEN(CCCD) = 12)  - UNIQUE CCCD  - NOT NULL DiaChi  - CHECK (LEN(SDT) = 10) |
| 7 | PhieuMuonTra | - Khóa chính MaPhieu  - Khóa ngoại TenDocGia tham chiếu đến bảng TacGia  - Khóa ngoại MaNhanVien tham chiếu đến bảng NhanVien  - NOT NULL NgayMuon  - NOT NULL NgayHetHan |
| 8 | CT\_MuonTra | - Khóa chính MaCuon  - Khóa ngoại MaPhieu tham chiếu đến bảng PhieuMuonTra  - CHECK TinhTrang IN (‘BT’, ‘Hư’, ‘Mất’)  - CHECK TienPhat >= 0  - CHECK SoNgayQuaHan >= 0  - CHECK ThanhToan IN (‘ĐT’, ‘CT’) |
| 9 | NhanVien | - Khóa chính MaNhanVien  - NOT NULL HoTen  - CHECK (LEN(CCCD) = 12)  - UNIQUE CCCD  - NOT NULL DiaChi  - CHECK (LEN(SDT) = 10) |
| 10 | DangNhap | -Khóa chính TenDangNhap  -NOL NULL MatKhau |

## **Cài đặt CSDL và các ràng buộc đơn giản**

|  |
| --- |
| -- Bảng NXB  CREATE TABLE NXB  (  MaNXB NVARCHAR(10) PRIMARY KEY,  TenNXB NVARCHAR(100) NOT NULL  );  GO |

|  |
| --- |
| -- Bảng TheLoai  CREATE TABLE TheLoai  (  MaTheLoai NVARCHAR(10) PRIMARY KEY,  TenTheLoai NVARCHAR(100) NOT NULL  );  GO |

|  |
| --- |
| -- Bảng TacGia  CREATE TABLE TacGia  (  MaTacGia NVARCHAR(10) PRIMARY KEY,  TenTacGia NVARCHAR(100) NOT NULL  );  GO |

|  |
| --- |
| -- Bảng DauSach  CREATE TABLE DauSach  (  MaSach NVARCHAR(10) PRIMARY KEY,  TenSach NVARCHAR(100) NOT NULL,  MaTacGia NVARCHAR(10) REFERENCES TacGia(MaTacGia) ON UPDATE CASCADE ON DELETE CASCADE,  MaNXB NVARCHAR(10) REFERENCES NXB(MaNXB) ON UPDATE CASCADE ON DELETE CASCADE,  MaTheLoai NVARCHAR(10) REFERENCES TheLoai(MaTheLoai) ON UPDATE CASCADE ON DELETE CASCADE,  Gia MONEY CHECK (Gia >= 0),  SoLuong INT CHECK (SoLuong >= 0)  );  GO |

|  |
| --- |
| -- Bảng CuonSach  CREATE TABLE CuonSach  (  MaCuon NVARCHAR(10) PRIMARY KEY,  MaSach NVARCHAR(10) REFERENCES DauSach(MaSach) ON UPDATE CASCADE ON DELETE CASCADE,  );  GO |

|  |
| --- |
| -- Bảng DocGia  CREATE TABLE DocGia  (  MaDocGia NVARCHAR(10) PRIMARY KEY,  TenDocGia NVARCHAR(50) NOT NULL,  GioiTinh NVARCHAR(4) CHECK (GioiTinh IN (N'Nam', N'Nữ')),  CCCD NVARCHAR(12) UNIQUE CHECK (LEN(CCCD) = 12),  DiaChi NVARCHAR(100) NOT NULL,  SDT NVARCHAR(10) CHECK (LEN(SDT) = 10),  Email NVARCHAR(50) CHECK (Email LIKE '%\_@\_%.\_%'),  )  GO |

|  |
| --- |
| -- Bảng NhanVien  CREATE TABLE NhanVien  (  MaNhanVien NVARCHAR(10) PRIMARY KEY,  HoTen NVARCHAR(50) NOT NULL,  CCCD NVARCHAR(12) UNIQUE CHECK (LEN(CCCD) = 12),  SDT NVARCHAR(10) CHECK (LEN(SDT) = 10),  DiaChi NVARCHAR(100) NOT NULL,  Email NVARCHAR(50)  )  GO |

|  |
| --- |
| -- Bảng PhieuMuonTra  CREATE TABLE PhieuMuonTra  (  MaPhieu NVARCHAR(10) PRIMARY KEY,  MaDocGia NVARCHAR(10) REFERENCES DocGia(MaDocGia) ON UPDATE CASCADE ON DELETE CASCADE,  MaNhanVien NVARCHAR(10) REFERENCES NhanVien(MaNhanVien) ON UPDATE CASCADE ON DELETE CASCADE,  NgayMuon DATE NOT NULL,  NgayHetHan AS DATEADD(DAY, 30, NgayMuon)  );  GO |

|  |
| --- |
| -- Bảng CT\_MuonTra  CREATE TABLE CT\_MuonTra  (  MaCuon NVARCHAR(10) PRIMARY KEY REFERENCES CuonSach(MaCuon),  MaPhieu NVARCHAR(10) REFERENCES PhieuMuonTra (MaPhieu) ON UPDATE CASCADE ON DELETE CASCADE,  NgayTra DATE,  TinhTrang NVARCHAR(4) CHECK (TinhTrang IN (N'BT', N'Hư', N'Mất')),  SoNgayQuaHan INT DEFAULT 0,  GhiChu NVARCHAR(200),  TienPhat MONEY CHECK (TienPhat >= 0),  ThanhToan NVARCHAR(2) CHECK (ThanhToan IN (N'ĐT', N'CT'))  )  GO |

|  |
| --- |
| -- Bảng SangTac  CREATE TABLE SangTac  (  MaSach NVARCHAR(10) REFERENCES DauSach(MaSach),  MaTacGia NVARCHAR(10) REFERENCES TacGia(MaTacGia),  PRIMARY KEY (MaSach, MaTacGia),  );  GO |

|  |
| --- |
| -- Bảng DauSach\_TheLoai  CREATE TABLE DauSach\_TheLoai  (  MaSach NVARCHAR(10) REFERENCES DauSach(MaSach),  MaTheLoai NVARCHAR(10) REFERENCES TheLoai(MaTheLoai),  PRIMARY KEY (MaSach, MaTheLoai)  );  GO |

|  |
| --- |
| --bảng DangNhap  CREATE TABLE DangNhap  (  TenDangNhap NVARCHAR(10) PRIMARY KEY REFERENCES NhanVien(MaNhanVien),  MatKhau NVARCHAR(50) NOT NULL,  CONSTRAINT FK\_DangNhap\_NhanVien FOREIGN KEY (TenDangNhap) REFERENCES NhanVien(MaNhanVien) ON UPDATE CASCADE ON DELETE CASCADE  ); |

## **Trigger**

|  |
| --- |
| -- 1. Trigger tự động thêm dữ liệu vào bảng CuonSach dựa theo SoLuong bảng DauSach  CREATE TRIGGER Trigger\_CapNhatSoLuongCuonSach  ON CuonSach  AFTER INSERT, DELETE  AS  BEGIN  -- Xử lý cho INSERT  IF EXISTS (SELECT \* FROM inserted)  BEGIN  UPDATE DauSach  SET SoLuong = SoLuong + (SELECT COUNT(MaCuon) FROM inserted WHERE inserted.MaSach = DauSach.MaSach)  FROM DauSach  WHERE EXISTS (SELECT \* FROM inserted WHERE inserted.MaSach = DauSach.MaSach);  END  -- Xử lý cho DELETE  IF EXISTS (SELECT \* FROM deleted)  BEGIN  UPDATE DauSach  SET SoLuong = SoLuong - (SELECT COUNT(MaCuon) FROM deleted WHERE deleted.MaSach = DauSach.MaSach)  FROM DauSach  WHERE EXISTS (SELECT \* FROM deleted WHERE deleted.MaSach = DauSach.MaSach);  END  END |

|  |
| --- |
| -- 2. Trigger cập nhật số ngày quá hạn và tính tiền phạt trong bảng CT\_MuonTra  CREATE OR ALTER TRIGGER Trigger\_CapNhatSoNgayQuaHanVaTienPhat  ON CT\_MuonTra  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @InsertedRows TABLE  (  MaCuon NVARCHAR(10) PRIMARY KEY,  MaPhieu NVARCHAR(10),  NgayTra DATE,  TinhTrang NVARCHAR(4)  );  INSERT INTO @InsertedRows (MaCuon, MaPhieu, NgayTra, TinhTrang)  SELECT MaCuon, MaPhieu, NgayTra, TinhTrang  FROM inserted;  UPDATE CT\_MuonTra  SET SoNgayQuaHan = DATEDIFF(DAY, PMT.NgayHetHan, IR.NgayTra)  FROM CT\_MuonTra  INNER JOIN @InsertedRows IR ON CT\_MuonTra.MaCuon = IR.MaCuon  INNER JOIN dbo.PhieuMuonTra PMT ON CT\_MuonTra.MaPhieu = PMT.MaPhieu;  UPDATE CT\_MuonTra  SET SoNgayQuaHan = 0  FROM CT\_MuonTra  INNER JOIN @InsertedRows IR ON CT\_MuonTra.MaCuon = IR.MaCuon  WHERE CT\_MuonTra.SoNgayQuaHan < 0;  UPDATE CT\_MuonTra  SET TienPhat = CT\_MuonTra.SoNgayQuaHan \* 1000 +  CASE  WHEN IR.TinhTrang = N'Mất' THEN 3 \* DS.Gia  WHEN IR.TinhTrang = N'Hư' THEN 0.5 \* DS.Gia  ELSE 0  END  FROM CT\_MuonTra AS CT\_MuonTra  JOIN CuonSach AS CS ON CT\_MuonTra.MaCuon = CS.MaCuon  JOIN DauSach AS DS ON CS.MaSach = DS.MaSach  INNER JOIN @InsertedRows IR ON CT\_MuonTra.MaCuon = IR.MaCuon;  END;  GO |

|  |
| --- |
| -- 3. Trigger kiểm tra số lượng sách mà độc giả mượn có vượt quá quy định không  CREATE TRIGGER trg\_KiemTraSoLuongMuonSach  ON CT\_MuonTra  AFTER INSERT  AS  BEGIN  DECLARE @MaDocGia NVARCHAR(10);  -- Lấy danh sách các độc giả mới mượn sách  SELECT @MaDocGia = MaDocGia  FROM PhieuMuonTra  WHERE MaPhieu IN (SELECT MaPhieu FROM inserted);  -- Kiểm tra số lượng mượn sách của độc giả  IF (SELECT COUNT(\*)  FROM CT\_MuonTra ctm  INNER JOIN PhieuMuonTra pmt ON ctm.MaPhieu = pmt.MaPhieu  WHERE pmt.MaDocGia = @MaDocGia) > 5  BEGIN  RAISERROR('Độc giả không được mượn quá 5 cuốn sách!', 16, 1);  ROLLBACK;  END  END;  GO |

|  |
| --- |
| -- 4. Trigger kiểm tra độc giả có đang mượn sách quá hạn không  CREATE TRIGGER trg\_KiemTraMuonQuaHan  ON CT\_MuonTra  AFTER INSERT  AS  BEGIN  DECLARE @MaPhieu NVARCHAR(10);  -- Lấy danh sách các độc giả mới mượn sách  SELECT @MaPhieu = MaPhieu  FROM inserted;  -- Kiểm tra sách mượn quá hạn  IF EXISTS (  SELECT 1  FROM PhieuMuonTra pm  INNER JOIN CT\_MuonTra ctm ON pm.MaPhieu = ctm.MaPhieu  WHERE ctm.ThanhToan = 'CT' AND ctm.NgayTra > pm.NgayHetHan AND pm.MaPhieu = @MaPhieu  )  BEGIN  RAISERROR('Độc giả có sách đang mượn quá hạn!', 16, 1);  ROLLBACK;  END  END;  GO |

## **Views**

|  |
| --- |
| -- 1. Xem chi tiết thông tin sách  CREATE OR ALTER VIEW view\_CTThongTinSach AS  SELECT  DS.MaSach AS MaSach,  DS.TenSach AS TenSach,  TG.TenTacGia AS TenTacGia,  NXB.TenNXB AS TenNXB,  TL.TenTheLoai AS TenTheLoai,  DS.Gia AS Gia,  DS.SoLuong AS SoLuong,  TG.MaTacGia AS MaTacGia,  NXB.MaNXB AS MaNXB,  TL.MaTheLoai AS MaTheLoai  FROM DauSach DS  JOIN TacGia TG ON DS.MaTacGia = TG.MaTacGia  JOIN NXB ON DS.MaNXB = NXB.MaNXB  JOIN TheLoai TL ON DS.MaTheLoai = TL.MaTheLoai  GO |

|  |
| --- |
| -- 2. Xem chi tiết thông tin Độc giả  CREATE OR ALTER VIEW view\_CTThongTinDocGia AS  SELECT  DG.MaDocGia AS MaDocGia,  DG.TenDocGia AS TenDocGia,  DG.GioiTinh AS GioiTinh,  DG.CCCD AS CCCD,  DG.DiaChi AS DiaChi,  DG.SDT AS SDT,  DG.Email AS Email  FROM DocGia DG  GO |

|  |
| --- |
| -- 3. Xem chi tiết thông tin Nhân Viên  CREATE OR ALTER VIEW view\_CTThongTinNV AS  SELECT  NV.MaNhanVien AS MaNhanVien,  NV.HoTen AS HoTen,  NV.CCCD AS CCCD,  NV.SDT AS SDT,  NV.DiaChi AS DiaChi,  NV.Email AS Email  FROM NhanVien NV  GO |

|  |
| --- |
| --4. Xem chi tiết thông tin phiếu mượn  CREATE OR ALTER VIEW view\_ThongTinPhieuMuon AS  SELECT  PMT.MaPhieu AS MaPhieu,  PMT.MaDocGia AS MaDocGia,  DG.TenDocGia AS TenDocGia,  PMT.MaNhanVien AS MaNhanVien,  NV.HoTen AS TenNhanVien,  PMT.NgayMuon AS NgayMuon,  PMT.NgayHetHan AS NgayHetHan  FROM PhieuMuonTra PMT  JOIN DocGia DG ON PMT.MaDocGia = DG.MaDocGia  JOIN NhanVien NV ON PMT.MaNhanVien = NV.MaNhanVien;  GO |

|  |
| --- |
| --5. Xem chi tiết thông tin phiếu trả (CT\_MuonTra)  CREATE VIEW view\_XemPhieuTra  AS SELECT \* FROM CT\_MuonTra  GO |

# **CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

* + - 1. **Kết nối cơ sở liệu**

|  |
| --- |
| public class SqlConnect  {  static public string connectionString = "Data Source=.;Initial Catalog=Nhom8\_QLTV;Integrated Security=True";  public SqlConnection con;  public SqlConnect()  {  con = new SqlConnection(connectionString);  }  public void OpenConnection()  {  if (con.State == ConnectionState.Closed)  {  con.Open();  }  }  public void CloseConnection()  {  if (con.State == ConnectionState.Open)  {  con.Close();  }  }  public DataTable ExecuteQuery(string query, SqlParameter[] parameters = null)  {  DataTable dt = new DataTable();  OpenConnection();  using (SqlCommand cmd = new SqlCommand(query, con))  {  if (parameters != null)  {  cmd.Parameters.AddRange(parameters);  }  using (SqlDataAdapter da = new SqlDataAdapter(cmd))  {  da.Fill(dt);  }  }  CloseConnection();  return dt;  }  } |

* + - 1. **Xem thông tin danh mục**
* **Danh mục Sách**

|  |
| --- |
| CREATE OR ALTER VIEW view\_CTThongTinSach AS  SELECT  DS.MaSach AS MaSach,  DS.TenSach AS TenSach,  TG.TenTacGia AS TenTacGia,  NXB.TenNXB AS TenNXB,  TL.TenTheLoai AS TenTheLoai,  DS.Gia AS Gia,  DS.SoLuong AS SoLuong,  TG.MaTacGia AS MaTacGia,  NXB.MaNXB AS MaNXB,  TL.MaTheLoai AS MaTheLoai  FROM DauSach DS  JOIN TacGia TG ON DS.MaTacGia = TG.MaTacGia  JOIN NXB ON DS.MaNXB = NXB.MaNXB  JOIN TheLoai TL ON DS.MaTheLoai = TL.MaTheLoai  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void XemSach()  {  try  {  sqlConnect.OpenConnection();  string sql = "SELECT \* FROM view\_CTThongTinSach";  dgv\_Sach.DataSource = sqlConnect.ExecuteQuery(sql);  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  } |

* **Danh mục Độc Giả**

|  |
| --- |
| CREATE OR ALTER VIEW view\_CTThongTinDocGia AS  SELECT  DG.MaDocGia AS MaDocGia,  DG.TenDocGia AS TenDocGia,  DG.GioiTinh AS GioiTinh,  DG.CCCD AS CCCD,  DG.DiaChi AS DiaChi,  DG.SDT AS SDT,  DG.Email AS Email  FROM DocGia DG  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void XemDocGia()  {  try  {  sqlConnect.OpenConnection();  string sql = "SELECT \* FROM view\_CTThongTinDocGia";  dgv\_DocGia.DataSource = sqlConnect.ExecuteQuery(sql);  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

* **Danh mục Nhân Viên**

|  |
| --- |
| CREATE OR ALTER VIEW view\_CTThongTinNV AS  SELECT  NV.MaNhanVien AS MaNhanVien,  NV.HoTen AS HoTen,  NV.CCCD AS CCCD,  NV.SDT AS SDT,  NV.DiaChi AS DiaChi,  NV.Email AS Email  FROM NhanVien NV  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void LoadNhanVien()  {  string sql = "SELECT \* FROM view\_CTThongTinNV";  dgv\_NhanVien.DataSource = sqlConnect.ExecuteQuery(sql);  }  public void XemNhanVien()  {  try  {  sqlConnect.OpenConnection();  LoadNhanVien();  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

* + - 1. **Quản lý sách**
  1. **Tìm kiếm chi tiết sách**
* **Theo “Tên sách”**

|  |
| --- |
| -- func1 Tìm sách theo tên sách  CREATE OR ALTER FUNCTION func\_TimSachTheoTenSach  (@MaSach nvarchar(100))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_CTThongTinSach  WHERE TenSach LIKE N'%'+@MaSach+'%'  )  GO |

* **Theo “Tên tác giả”**

|  |
| --- |
| -- func2 Tìm sách theo tác giả  CREATE OR ALTER FUNCTION func\_TimSachTheoTenTacGia  (@TenTacGia nvarchar(100))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_CTThongTinSach  WHERE TenTacGia LIKE N'%'+@TenTacGia+'%'  )  GO |

* **Theo “Tên thể loại”**

|  |
| --- |
| -- func3 Tìm sách theo thể loại  CREATE OR ALTER FUNCTION func\_TimSachTenTheLoai  (@TenTheLoai nvarchar(100))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_CTThongTinSach  WHERE TenTheLoai LIKE N'%'+@TenTheLoai+'%'  )  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void TimKiemSach()  {  string tuKhoa = txt\_DauSach\_TimKiem.Text.Trim();  if (string.IsNullOrEmpty(tuKhoa))  {  MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (rbt\_DauSach\_TenSach.Checked)  {  string query = "SELECT \* FROM func\_TimSachTheoTenSach(@TenSach)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TenSach", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_Sach.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_DauSach\_TenTacGia.Checked)  {  string query = "SELECT \* FROM func\_TimSachTheoTenTacGia(@TenTacGia)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TenTacGia", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_Sach.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_DauSach\_TheLoai.Checked)  {  string query = "SELECT \* FROM func\_TimSachTenTheLoai(@TenTheLoai)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TenTheLoai", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_Sach.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  MessageBox.Show("Không tìm thấy nhân viên phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

* 1. **Tạo sách mới thêm vào bảng chi tiết sách**

|  |
| --- |
| -- proc1 Thêm sách  CREATE OR ALTER PROCEDURE proc\_ThemSach  @MaSach nvarchar(10),  @TenSach nvarchar(100),  @MaTacGia nvarchar(10),  @NXB nvarchar(10),  @TheLoai nvarchar(10),  @DonGia money,  @SoLuong int  AS  BEGIN  IF EXISTS (SELECT 1 FROM DauSach WHERE MaSach = @MaSach)  BEGIN  RAISERROR(N'Mã sách đã tồn tại!', 16, 1);  RETURN;  END  INSERT INTO DauSach (MaSach, TenSach, MaTacGia, MaNXB, MaTheLoai, Gia, SoLuong)  VALUES (@MaSach, @TenSach, @MaTacGia, @NXB, @TheLoai, @DonGia, @SoLuong)  END  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void ThemSach()  {  try  {  string maSach = txt\_DauSach\_MaSach.Text.Trim();  string tenSach = txt\_DauSach\_TenSach.Text.Trim();  string maTacGia = txt\_DauSach\_MaTacGia.Text.Trim();  string maNXB = txt\_DauSach\_MaNXB.Text.Trim();  string maTheLoai = txt\_DauSach\_MaTheLoai.Text.Trim();  string gia = txt\_DauSach\_DonGia.Text.Trim();  string soLuong = txt\_DauSach\_SoLuong.Text.Trim();  if (string.IsNullOrEmpty(maSach) || string.IsNullOrEmpty(tenSach) || string.IsNullOrEmpty(maTacGia) ||  string.IsNullOrEmpty(maNXB) || string.IsNullOrEmpty(maTheLoai))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThemSach", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txt\_DauSach\_MaSach.Text.Trim();  cmd.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value = txt\_DauSach\_TenSach.Text.Trim();  cmd.Parameters.Add("@MaTacGia", SqlDbType.NVarChar).Value = txt\_DauSach\_MaTacGia.Text.Trim();  cmd.Parameters.Add("@NXB", SqlDbType.NVarChar).Value = txt\_DauSach\_MaNXB.Text.Trim();  cmd.Parameters.Add("@TheLoai", SqlDbType.NVarChar).Value = txt\_DauSach\_MaTheLoai.Text.Trim();  cmd.Parameters.Add("@DonGia", SqlDbType.NVarChar).Value = txt\_DauSach\_DonGia.Text.Trim();  cmd.Parameters.Add("@SoLuong", SqlDbType.NVarChar).Value = txt\_DauSach\_SoLuong.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công");  LoadSach();  }  else  {  MessageBox.Show("Thêm thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Chỉnh sửa thông tin sách**

|  |
| --- |
| CREATE OR ALTER PROCEDURE proc\_SuaSach  (  @MaSach nvarchar(10),  @TenSach nvarchar(100),  @MaTacGia nvarchar(10),  @NXB nvarchar(10),  @TheLoai nvarchar(10),  @DonGia money,  @SoLuong int  )  AS  BEGIN  UPDATE DauSach  SET TenSach = @TenSach,  MaTacGia = @MaTacGia,  MaNXB = @NXB,  MaTheLoai = @TheLoai,  Gia = @DonGia,  SoLuong = @SoLuong  WHERE MaSach = @MaSach  SELECT COUNT(\*) AS SoLuong  FROM DauSach  WHERE MaSach = @MaSach  END  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void SuaSach()  {  try  {  string maSach = txt\_DauSach\_MaSach.Text.Trim();  string tenSach = txt\_DauSach\_TenSach.Text.Trim();  string maTacGia = txt\_DauSach\_MaTacGia.Text.Trim();  string maNXB = txt\_DauSach\_MaNXB.Text.Trim();  string maTheLoai = txt\_DauSach\_MaTheLoai.Text.Trim();  string gia = txt\_DauSach\_DonGia.Text.Trim();  string soLuong = txt\_DauSach\_SoLuong.Text.Trim();  if (string.IsNullOrEmpty(maSach) || string.IsNullOrEmpty(tenSach) || string.IsNullOrEmpty(maTacGia) ||  string.IsNullOrEmpty(maNXB) || string.IsNullOrEmpty(maTheLoai))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_SuaSach", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txt\_DauSach\_MaSach.Text.Trim();  cmd.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value = txt\_DauSach\_TenSach.Text.Trim();  cmd.Parameters.Add("@MaTacGia", SqlDbType.NVarChar).Value = txt\_DauSach\_MaTacGia.Text.Trim();  cmd.Parameters.Add("@NXB", SqlDbType.NVarChar).Value = txt\_DauSach\_MaNXB.Text.Trim();  cmd.Parameters.Add("@TheLoai", SqlDbType.NVarChar).Value = txt\_DauSach\_MaTheLoai.Text.Trim();  cmd.Parameters.Add("@DonGia", SqlDbType.NVarChar).Value = txt\_DauSach\_DonGia.Text.Trim();  cmd.Parameters.Add("@SoLuong", SqlDbType.NVarChar).Value = txt\_DauSach\_SoLuong.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận sửa?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Sửa thành công");  LoadSach();  }  else  {  MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Xóa sách**

|  |
| --- |
| -- proc3 Xóa sách  CREATE PROCEDURE proc\_XoaSach  @MaSach NVARCHAR(10)  AS  BEGIN  IF NOT EXISTS (SELECT 1 FROM DauSach WHERE MaSach = @MaSach)  BEGIN  RAISERROR(N'Mã sách không tồn tại!', 16, 1);  RETURN;  END  DELETE FROM DauSach WHERE MaSach = @MaSach;  END  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void XoaSach()  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_XoaSach", sqlConnect.con);  cmd.Parameters.Add("@MaSach", SqlDbType.NVarChar).Value = txt\_DauSach\_MaSach.Text.Trim();  cmd.CommandType = CommandType.StoredProcedure;  DialogResult result = MessageBox.Show("Xác nhận xóa", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Xóa thành công");  LoadSach();  }  }  else  {  MessageBox.Show("Xóa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* + - 1. **Quản lí độc giả**
  1. **Thêm độc giả mới**

|  |
| --- |
| -- proc4 Thêm độc giả  CREATE OR ALTER PROCEDURE proc\_ThemDocGia  @MaDocGia NVARCHAR(10),  @TenDocGia NVARCHAR(50),  @GioiTinh NVARCHAR(4),  @CCCD NVARCHAR(12),  @DiaChi NVARCHAR(100),  @SDT NVARCHAR(10),  @Email NVARCHAR(50)  AS  BEGIN  IF EXISTS (SELECT 1 FROM DocGia WHERE MaDocGia = @MaDocGia)  BEGIN  RAISERROR(N'Mã độc giả đã tồn tại!', 16, 1);  RETURN;  END  INSERT INTO DocGia (MaDocGia, TenDocGia, GioiTinh, CCCD, DiaChi, SDT, Email)  VALUES (@MaDocGia, @TenDocGia, @GioiTinh, @CCCD, @DiaChi, @SDT, @Email);  END  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void ThemDocGia()  {  try  {  string maDocGia = txt\_DocGia\_MaDocGia.Text.Trim();  string tenDocGia = txt\_DocGia\_TenDocGia.Text.Trim();  string gioiTinh = cbo\_DocGia\_GioiTinh.Text.Trim();  string cccd = txt\_DocGia\_CCCD.Text.Trim();  string diaChi = txt\_DocGia\_DiaChi.Text.Trim();  string sdt = txt\_DocGia\_SDT.Text.Trim();  if (string.IsNullOrEmpty(maDocGia) || string.IsNullOrEmpty(tenDocGia) || string.IsNullOrEmpty(gioiTinh) ||  string.IsNullOrEmpty(cccd) || string.IsNullOrEmpty(diaChi) || string.IsNullOrEmpty(sdt))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThemDocGia", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaDocGia", SqlDbType.NVarChar).Value = txt\_DocGia\_MaDocGia.Text.Trim();  cmd.Parameters.Add("@TenDocGia", SqlDbType.NVarChar).Value = txt\_DocGia\_TenDocGia.Text.Trim();  cmd.Parameters.Add("@GioiTinh", SqlDbType.NVarChar).Value = cbo\_DocGia\_GioiTinh.Text.Trim();  cmd.Parameters.Add("@CCCD", SqlDbType.NVarChar).Value = txt\_DocGia\_CCCD.Text.Trim();  cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txt\_DocGia\_DiaChi.Text.Trim();  cmd.Parameters.Add("@SDT", SqlDbType.NVarChar).Value = txt\_DocGia\_SDT.Text.Trim();  cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = txt\_DocGia\_Email.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công");  LoadDocGia();  }  else  {  MessageBox.Show("Thêm thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### **Sửa thông tin độc giả**

|  |
| --- |
| -- proc5 Sửa thông tin độc giả--  CREATE OR ALTER PROCEDURE proc\_SuaDocGia  @MaDocGia NVARCHAR(10),  @TenDocGia NVARCHAR(50),  @GioiTinh NVARCHAR(4),  @CCCD NVARCHAR(12),  @DiaChi NVARCHAR(100),  @SDT NVARCHAR(10),  @Email NVARCHAR(50)  AS  BEGIN  UPDATE DocGia  SET TenDocGia = @TenDocGia,  GioiTinh = @GioiTinh,  CCCD = @CCCD,  DiaChi = @DiaChi,  SDT = @SDT,  Email = @Email  WHERE MaDocGia = @MaDocGia;  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void SuaDocGia()  {  try  {  string maDocGia = txt\_DocGia\_MaDocGia.Text.Trim();  string tenDocGia = txt\_DocGia\_TenDocGia.Text.Trim();  string gioiTinh = cbo\_DocGia\_GioiTinh.Text.Trim();  string cccd = txt\_DocGia\_CCCD.Text.Trim();  string diaChi = txt\_DocGia\_DiaChi.Text.Trim();  string sdt = txt\_DocGia\_SDT.Text.Trim();  if (string.IsNullOrEmpty(maDocGia) || string.IsNullOrEmpty(tenDocGia) || string.IsNullOrEmpty(gioiTinh) ||  string.IsNullOrEmpty(cccd) || string.IsNullOrEmpty(diaChi) || string.IsNullOrEmpty(sdt))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_SuaDocGia", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaDocGia", SqlDbType.NVarChar).Value = txt\_DocGia\_MaDocGia.Text.Trim();  cmd.Parameters.Add("@TenDocGia", SqlDbType.NVarChar).Value = txt\_DocGia\_TenDocGia.Text.Trim();  cmd.Parameters.Add("@GioiTinh", SqlDbType.NVarChar).Value = cbo\_DocGia\_GioiTinh.Text.Trim();  cmd.Parameters.Add("@CCCD", SqlDbType.NVarChar).Value = txt\_DocGia\_CCCD.Text.Trim();  cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txt\_DocGia\_DiaChi.Text.Trim();  cmd.Parameters.Add("@SDT", SqlDbType.NVarChar).Value = txt\_DocGia\_SDT.Text.Trim();  cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = txt\_DocGia\_Email.Text.Trim();  if (cmd.ExecuteNonQuery() > 0)  {  DialogResult result = MessageBox.Show("Xác nhận sửa?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  MessageBox.Show("Sửa thành công");  LoadDocGia();  }  }  else  {  MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### **Xóa độc giả**

|  |
| --- |
| -- proc6 Xóa độc giả--  CREATE OR ALTER PROCEDURE proc\_XoaDocGia  @MaDocGia NVARCHAR(10)  AS  BEGIN  -- Kiểm tra xem độc giả có tồn tại không  IF NOT EXISTS (SELECT 1 FROM DocGia WHERE MaDocGia = @MaDocGia)  BEGIN  RAISERROR(N'Độc giả không tồn tại!', 16, 1);  RETURN;  END  -- Xóa độc giả từ bảng DocGia  DELETE FROM DocGia WHERE MaDocGia = @MaDocGia;  END |

* **Thực thi trên C#**

|  |
| --- |
| public void XoaDocGia()  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_XoaDocGia", sqlConnect.con);  cmd.Parameters.Add("@MaDocGia", SqlDbType.NVarChar).Value = txt\_DocGia\_MaDocGia.Text.Trim();  cmd.CommandType = CommandType.StoredProcedure;  DialogResult result = MessageBox.Show("Xác nhận xóa", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Xóa thành công");  LoadDocGia();  }  }  else  {  MessageBox.Show("Xóa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### **Tìm kiếm độc giả**

* **Theo “Mã độc giả”**

|  |
| --- |
| -- func4 Tìm độc giả theo mã  CREATE OR ALTER FUNCTION func\_TimDocGiaTheoMa  (@MaDocGia NVARCHAR(10))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM DocGia  WHERE MaDocGia = @MaDocGia  )  GO |

* **Theo “Tên độc giả”**

|  |
| --- |
| --func5 Tìm độc giả theo tên  CREATE OR ALTER FUNCTION func\_TimDocGiaTheoTen  (@TenDocGia NVARCHAR(50))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM DocGia  WHERE TenDocGia LIKE '%' + @TenDocGia + '%'  )  GO |

* **Theo “Số điện thoại”**

|  |
| --- |
| -- func6 Tìm độc giả theo SĐT  CREATE OR ALTER FUNCTION func\_TimDocGiaTheoSDT  (@SDT NVARCHAR(10))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM DocGia  WHERE SDT = @SDT  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void TimKiemDocGia()  {  string tuKhoa = txt\_DocGia\_TimKiem.Text.Trim();  if (string.IsNullOrEmpty(tuKhoa))  {  MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (rbt\_DocGia\_MaDocGia.Checked)  {  string query = "SELECT \* FROM dbo.func\_TimDocGiaTheoMa(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_DocGia.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy độc giả phù hợp: " + tuKhoa, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_DocGia\_TenDocGia.Checked)  {  string query = "SELECT \* FROM dbo.func\_TimDocGiaTheoTen(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_DocGia.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy độc giả phù hợp: " + tuKhoa, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_DocGia\_SDT.Checked)  {  string query = "SELECT \* FROM dbo.func\_TimDocGiaTheoSDT(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_DocGia.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy độc giả phù hợp: " + tuKhoa, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  MessageBox.Show("Không tìm thấy độc giả phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

1. **Quản lý phiếu mượn trả**
   1. **Thêm phiếu mượn**

|  |
| --- |
| -- proc7 Thêm phiếu mượn  CREATE OR ALTER PROCEDURE proc\_ThemPhieuMuon  @MaPhieu nvarchar(10),  @MaDocGia nvarchar(10),  @MaNhanVien nvarchar(10),  @NgayMuon date  AS  BEGIN  IF EXISTS (SELECT 1 FROM PhieuMuonTra WHERE MaPhieu =@MaPhieu)  BEGIN  RAISERROR(N'Mã phiếu đã tồn tại!', 16, 1);  RETURN;  END  INSERT INTO PhieuMuonTra (MaPhieu, MaDocGia, MaNhanVien, NgayMuon)  VALUES (@MaPhieu, @MaDocGia, @MaNhanVien, @NgayMuon)  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void ThemPhieuMuon()  {  try  {  string maPhieu = txt\_PhieuMuon\_MaPhieu.Text.Trim();  string maDocGia = txt\_PhieuMuon\_MaDocGia.Text.Trim();  string maMaNhanVien = txt\_PhieuMuon\_MaNhanVien.Text.Trim();    if (string.IsNullOrEmpty(maPhieu) || string.IsNullOrEmpty(maDocGia) || string.IsNullOrEmpty(maMaNhanVien))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThemPhieuMuon", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaPhieu", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaPhieu.Text.Trim();  cmd.Parameters.Add("@MaDocGia", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaDocGia.Text.Trim();  cmd.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaNhanVien.Text.Trim();  cmd.Parameters.Add("@NgayMuon", SqlDbType.Date).Value = dtp\_PhieuMuon\_NgayMuon.Value;  dtp\_PhieuMuon\_NgayHetHan.Value = dtp\_PhieuMuon\_NgayMuon.Value.AddDays(30);  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công");  LoadPhieuMuon();  }  else  {  MessageBox.Show("Thêm thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Cập nhật phiếu mượn**

|  |
| --- |
| -- proc8 Cập nhật phiếu mượn  CREATE OR ALTER PROCEDURE proc\_SuaPhieuMuon  @MaPhieu nvarchar(1000),  @MaDocGia nvarchar(1000),  @MaNhanVien nvarchar(1000),  @NgayMuon date  AS  BEGIN  UPDATE PhieuMuonTra SET  MaPhieu = @MaPhieu,  MaDocGia = @MaDocGia,  MaNhanVien = @MaNhanVien,  NgayMuon = @NgayMuon  WHERE MaPhieu = @MaPhieu  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void SuaPhieuMuon()  {  try  {  string maPhieu = txt\_PhieuMuon\_MaPhieu.Text.Trim();  string maDocGia = txt\_PhieuMuon\_MaDocGia.Text.Trim();  string maMaNhanVien = txt\_PhieuMuon\_MaNhanVien.Text.Trim();    if (string.IsNullOrEmpty(maPhieu) || string.IsNullOrEmpty(maDocGia) || string.IsNullOrEmpty(maMaNhanVien) )  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_SuaPhieuMuon", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaPhieu", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaPhieu.Text.Trim();  cmd.Parameters.Add("@MaDocGia", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaDocGia.Text.Trim();  cmd.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txt\_PhieuMuon\_MaNhanVien.Text.Trim();  cmd.Parameters.Add("@NgayMuon", SqlDbType.Date).Value = dtp\_PhieuMuon\_NgayMuon.Value;  dtp\_PhieuMuon\_NgayHetHan.Value = dtp\_PhieuMuon\_NgayMuon.Value.AddDays(30);  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Sửa thành công");  LoadPhieuMuon();  }  else  {  MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Xem phiếu mượn**

|  |
| --- |
| --4. Xem chi tiết thông tin phiếu mượn  CREATE OR ALTER VIEW view\_ThongTinPhieuMuon AS  SELECT  PMT.MaPhieu AS MaPhieu,  PMT.MaDocGia AS MaDocGia,  DG.TenDocGia AS TenDocGia,  PMT.MaNhanVien AS MaNhanVien,  NV.HoTen AS TenNhanVien,  PMT.NgayMuon AS NgayMuon,  PMT.NgayHetHan AS NgayHetHan  FROM PhieuMuonTra PMT  JOIN DocGia DG ON PMT.MaDocGia = DG.MaDocGia  JOIN NhanVien NV ON PMT.MaNhanVien = NV.MaNhanVien;  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void LoadPhieuMuon()  {  string sql = "SELECT \* FROM view\_ThongTinPhieuMuon";  dgv\_PhieuMuon.DataSource = sqlConnect.ExecuteQuery(sql);  }  public void XemPhieuMuon()  {  try  {  sqlConnect.OpenConnection();  LoadPhieuMuon();  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

* 1. **Tìm kiếm phiếu mượn**
* **Theo “Mã phiếu”**

|  |
| --- |
| -- func7 Tìm phiếu mượn theo mã  CREATE FUNCTION func\_TimPhieuMuonTheoMaPhieu  (@MaPhieu NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_ThongTinPhieuMuon  WHERE MaPhieu = @MaPhieu  )  GO |

* **Theo “Tên độc giả”**

|  |
| --- |
| -- func8 Tìm phiếu mượn theo tên độc giả  CREATE FUNCTION func\_TimPhieuMuonTheoTenDocGia  (@TenDocGia NVARCHAR(50))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_ThongTinPhieuMuon  WHERE TenDocGia LIKE N'%'+@TenDocGia+'%'  )  GO |

* **Theo “Tên nhân viên”**

|  |
| --- |
| -- func9 Tìm phiếu mượn theo tên nhân viên  CREATE FUNCTION func\_TimPhieuMuonTheoTenNhanVien  (@TenNhanVien NVARCHAR(50))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_ThongTinPhieuMuon  WHERE TenNhanVien LIKE N'%'+@TenNhanVien+'%'  )  GO |

* **Theo “Ngày mượn”**

|  |
| --- |
| -- func10 Tìm phiếu mượn theo ngày mượn  CREATE FUNCTION func\_TimPhieuMuonTheoNgayMuon  (@NgayMuon DATE)  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM view\_ThongTinPhieuMuon  WHERE NgayMuon = @NgayMuon  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void TimKiemPhieuMuon()  {  if (rbt\_PhieuMuon\_NgayMuon.Checked)  {  DateTime date = dtp\_PhieuMuon\_TimKiem.Value;  string query = "SELECT \* FROM func\_TimPhieuMuonTheoNgayMuon(@NgayMuon)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@NgayMuon", date));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuMuon.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  string tuKhoa = txt\_PhieuMuon\_TimKiem.Text.Trim();  if (string.IsNullOrEmpty(tuKhoa))  {  MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (rbt\_PhieuMuon\_MaPhieu.Checked)  {  string query = "SELECT \* FROM func\_TimPhieuMuonTheoMaPhieu(@MaPhieu)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@MaPhieu", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuMuon.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_PhieuMuon\_TenDocGia.Checked)  {  string query = "SELECT \* FROM func\_TimPhieuMuonTheoTenDocGia(@TenDocGia)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TenDocGia", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuMuon.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_PhieuMuon\_TenNhanVien.Checked)  {  string query = "SELECT \* FROM func\_TimPhieuMuonTheoTenNhanVien(@TenNhanVien)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TenNhanVien", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuMuon.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  MessageBox.Show("Không tìm thấy nhân viên phù hợp!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  } |

1. **Quản lý phiếu trả (Chi tiết phiếu mượn trả)**
   1. **Thêm phiếu trả**

|  |
| --- |
| -- proc7 Thêm phiếu mượn  CREATE OR ALTER PROCEDURE proc\_ThemPhieuMuon  @MaPhieu nvarchar(10),  @MaDocGia nvarchar(10),  @MaNhanVien nvarchar(10),  @NgayMuon date  AS  BEGIN  IF EXISTS (SELECT 1 FROM PhieuMuonTra WHERE MaPhieu =@MaPhieu)  BEGIN  RAISERROR(N'Mã phiếu đã tồn tại!', 16, 1);  RETURN;  END  INSERT INTO PhieuMuonTra (MaPhieu, MaDocGia, MaNhanVien, NgayMuon)  VALUES (@MaPhieu, @MaDocGia, @MaNhanVien, @NgayMuon)  END  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void ThemPhieuTra()  {  try  {  string maPhieu = txt\_PhieuTra\_MaPhieu.Text.Trim();  string maCuon = txt\_PhieuTra\_MaCuon.Text.Trim();  DateTime ngayTra = dtp\_PhieuTra\_NgayTra.Value;  DateTime ngayHetHan = dtp\_PhieuMuon\_NgayHetHan.Value;  string tinhTrang=cmb\_PhieuTra\_TinhTrang.Text.Trim();  string ghiChu=txt\_PhieuTra\_GhiChu.Text.Trim();  string thanhToan=txt\_PhieuTra\_ThanhToan.Text.Trim();  int soNgayQuaHan = (int)(ngayTra - ngayHetHan).TotalDays;  if (string.IsNullOrEmpty(maPhieu) || string.IsNullOrEmpty(maCuon) || string.IsNullOrEmpty(tinhTrang)|| string.IsNullOrEmpty(ghiChu) || string.IsNullOrEmpty(thanhToan))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThemPhieuTra", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaPhieu", SqlDbType.NVarChar).Value = txt\_PhieuTra\_MaPhieu.Text.Trim();  cmd.Parameters.Add("@MaCuon", SqlDbType.NVarChar).Value = txt\_PhieuTra\_MaCuon.Text.Trim();  cmd.Parameters.Add("@NgayTra", SqlDbType.Date).Value = dtp\_PhieuTra\_NgayTra.Value;  cmd.Parameters.Add("@TinhTrang", SqlDbType.NVarChar).Value = cmb\_PhieuTra\_TinhTrang.Text.Trim();  cmd.Parameters.Add("@GhiChu", SqlDbType.NVarChar).Value=txt\_PhieuTra\_GhiChu.Text.Trim();  cmd.Parameters.Add("@ThanhToan",SqlDbType.NVarChar).Value=txt\_PhieuTra\_ThanhToan.Text.Trim();  txt\_PhieuTra\_SoNgayQuaHan.Text = soNgayQuaHan.ToString();  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công");  LoadPhieuTra();  }  else  {  MessageBox.Show("Thêm thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Cập nhật Phiếu Trả**

|  |
| --- |
| -- pro10 Sửa phiếu trả  CREATE OR ALTER PROCEDURE proc\_SuaPhieuTra  (  @MaCuon NVARCHAR(10),  @MaPhieu NVARCHAR(10),  @NgayTra DATE,  @TinhTrang NVARCHAR(10),  @GhiChu NVARCHAR(200),  @ThanhToan NVARCHAR(2)  )  AS  BEGIN  UPDATE CT\_MuonTra  SET  MaCuon = @MaCuon,  MaPhieu = @MaPhieu,  NgayTra = @NgayTra,  TinhTrang = @TinhTrang,  GhiChu = @GhiChu,  ThanhToan = @ThanhToan  WHERE  MaCuon = @MaCuon  RETURN  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void SuaPhieuTra()  {  try  {  string maPhieu = txt\_PhieuTra\_MaPhieu.Text.Trim();  string maCuon = txt\_PhieuTra\_MaCuon.Text.Trim();  DateTime ngayTra = dtp\_PhieuTra\_NgayTra.Value;  DateTime ngayHetHan = dtp\_PhieuMuon\_NgayHetHan.Value;  string tinhTrang = cmb\_PhieuTra\_TinhTrang.Text.Trim();  string ghiChu = txt\_PhieuTra\_GhiChu.Text.Trim();  string thanhToan = txt\_PhieuTra\_ThanhToan.Text.Trim();  int soNgayQuaHan = (int)(ngayTra - ngayHetHan).TotalDays;  if (string.IsNullOrEmpty(maPhieu) || string.IsNullOrEmpty(maCuon) || string.IsNullOrEmpty(tinhTrang) || string.IsNullOrEmpty(ghiChu) || string.IsNullOrEmpty(thanhToan))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_SuaPhieuTra", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaPhieu", SqlDbType.NVarChar).Value = txt\_PhieuTra\_MaPhieu.Text.Trim();  cmd.Parameters.Add("@MaCuon", SqlDbType.NVarChar).Value = txt\_PhieuTra\_MaCuon.Text.Trim();  cmd.Parameters.Add("@NgayTra", SqlDbType.Date).Value = dtp\_PhieuTra\_NgayTra.Value;  cmd.Parameters.Add("@TinhTrang", SqlDbType.NVarChar).Value = cmb\_PhieuTra\_TinhTrang.Text.Trim();  cmd.Parameters.Add("@GhiChu", SqlDbType.NVarChar).Value = txt\_PhieuTra\_GhiChu.Text.Trim();  cmd.Parameters.Add("@ThanhToan", SqlDbType.NVarChar).Value = txt\_PhieuTra\_ThanhToan.Text.Trim();  txt\_PhieuTra\_SoNgayQuaHan.Text = soNgayQuaHan.ToString();  DialogResult result = MessageBox.Show("Xác nhận sửa?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Sửa thành công");  LoadPhieuMuon();  }  else  {  MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Xem phiếu Trả**

|  |
| --- |
| --5. Xem chi tiết thông tin phiếu trả (CT\_MuonTra)  CREATE VIEW view\_XemPhieuTra  AS SELECT \* FROM CT\_MuonTra  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void LoadPhieuTra()  {  string sql = "SELECT \* FROM CT\_MuonTra";  dgv\_PhieuTra.DataSource = sqlConnect.ExecuteQuery(sql);  }  public void XemPhieuTra()  {  try  {  sqlConnect.OpenConnection();  LoadPhieuTra();  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

* 1. **Tìm kiếm phiếu trả**
* **Theo “Mã phiếu”**

|  |
| --- |
| -- func12 Tìm phiếu trả theo mã phiếu  CREATE FUNCTION func\_TimPhieuTraTheoMaPhieu  (@MaPhieu NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM CT\_MuonTra  WHERE MaPhieu = @MaPhieu  )  GO |

* **Theo “Mã cuốn”**

|  |
| --- |
| -- func11 Tìm phiếu trả theo mã cuốn  CREATE FUNCTION func\_TimPhieuTraTheoMaCuon  (@MaCuon NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM CT\_MuonTra  WHERE MaCuon = @MaCuon  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void TimKiemPhieuTra()  {  string tuKhoa = txt\_PhieuTra\_TimKiem.Text.Trim();  if (string.IsNullOrEmpty(tuKhoa))  {  MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (rbt\_PhieuTra\_MaCuon.Checked)  {  string query = "SELECT \* FROM func\_TimPhieuTraTheoMaCuon(@MaCuon)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@MaCuon", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuTra.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  if (rbt\_PhieuTra\_MaPhieu.Checked)  {  string query = "SELECT \* FROM func\_TimPhieuTraTheoMaPhieu(@MaPhieu)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@MaPhieu", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_PhieuTra.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }    }  } |

* 1. **Xóa phiếu trả**

|  |
| --- |
| -- proc11 Xóa phiếu trả  CREATE OR ALTER PROCEDURE proc\_XoaPhieuTra (  @MaCuon NVARCHAR(10)  )  AS  BEGIN  IF NOT EXISTS (SELECT 1 FROM CT\_MuonTra WHERE MaCuon = @MaCuon)  BEGIN  RAISERROR(N'Không tồn tại',16,1);  RETURN  DELETE FROM CT\_MuonTra  WHERE MaCuon = @MaCuon;  END  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void XoaPhieuTra()  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_XoaPhieuTra", sqlConnect.con);  cmd.Parameters.Add("@MaCuon", SqlDbType.NVarChar).Value = txt\_PhieuTra\_MaCuon.Text.Trim();  cmd.CommandType = CommandType.StoredProcedure;  DialogResult result = MessageBox.Show("Xác nhận xóa", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Xóa thành công");  LoadPhieuTra();  }  }  else  {  MessageBox.Show("Xóa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

1. **Quản lý nhân viên**
   1. **Thêm nhân viên**

|  |
| --- |
| -- proc12 Thêm nhân viên  CREATE OR ALTER PROCEDURE proc\_ThemNhanVien  @MaNhanVien NVARCHAR(10),  @HoTen NVARCHAR(50),  @CCCD NVARCHAR(12),  @DiaChi NVARCHAR(100),  @SDT NVARCHAR(10),  @Email NVARCHAR(50)  AS  BEGIN  IF EXISTS (SELECT 1 FROM NhanVien WHERE MaNhanVien = @MaNhanVien)  BEGIN  RAISERROR(N'Mã Nhân viên đã tồn tại!', 16, 1);  RETURN;  END  INSERT INTO NhanVien (MaNhanVien, HoTen, CCCD, DiaChi, SDT, Email)  VALUES (@MaNhanVien, @HoTen, @CCCD, @DiaChi, @SDT, @Email);  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void ThemNhanVien()  {  try  {  string maNhanVien = txt\_NhanVien\_MaNhanVien.Text.Trim();  string hoTen = txt\_NhanVien\_HoTen.Text.Trim();  string cccd = txt\_NhanVien\_CCCD.Text.Trim();  string diaChi = txt\_NhanVien\_DiaChi.Text.Trim();  string sdt = txt\_NhanVien\_SDT.Text.Trim();  if (string.IsNullOrEmpty(maNhanVien) || string.IsNullOrEmpty(hoTen) ||  string.IsNullOrEmpty(cccd) || string.IsNullOrEmpty(diaChi) || string.IsNullOrEmpty(sdt))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThemNhanVien", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txt\_NhanVien\_MaNhanVien.Text.Trim();  cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txt\_NhanVien\_HoTen.Text.Trim();  cmd.Parameters.Add("@CCCD", SqlDbType.NVarChar).Value = txt\_NhanVien\_CCCD.Text.Trim();  cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txt\_NhanVien\_DiaChi.Text.Trim();  cmd.Parameters.Add("@SDT", SqlDbType.NVarChar).Value = txt\_NhanVien\_SDT.Text.Trim();  cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = txt\_NhanVien\_Email.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận thêm?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm thành công");  LoadNhanVien();  }  else  {  MessageBox.Show("Thêm thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Cập nhật nhân viên**

|  |
| --- |
| -- proc13 Sửa nhân viên  CREATE OR ALTER PROCEDURE proc\_SuaNhanVien  @MaNhanVien NVARCHAR(10),  @HoTen NVARCHAR(50),  @CCCD NVARCHAR(12),  @DiaChi NVARCHAR(100),  @SDT NVARCHAR(10),  @Email NVARCHAR(50)  AS  BEGIN  UPDATE NhanVien  SET HoTen = @HoTen,  CCCD = @CCCD,  DiaChi = @DiaChi,  SDT = @SDT,  Email = @Email  WHERE MaNhanVien = @MaNhanVien;  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void SuaNhanVien()  {  try  {  string maNhanVien = txt\_NhanVien\_MaNhanVien.Text.Trim();  string hoTen = txt\_NhanVien\_HoTen.Text.Trim();  string cccd = txt\_NhanVien\_CCCD.Text.Trim();  string diaChi = txt\_NhanVien\_DiaChi.Text.Trim();  string sdt = txt\_NhanVien\_SDT.Text.Trim();  if (string.IsNullOrEmpty(maNhanVien) || string.IsNullOrEmpty(hoTen) ||  string.IsNullOrEmpty(cccd) || string.IsNullOrEmpty(diaChi) || string.IsNullOrEmpty(sdt))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_SuaNhanVien", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txt\_NhanVien\_MaNhanVien.Text.Trim();  cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = txt\_NhanVien\_HoTen.Text.Trim();  cmd.Parameters.Add("@CCCD", SqlDbType.NVarChar).Value = txt\_NhanVien\_CCCD.Text.Trim();  cmd.Parameters.Add("@DiaChi", SqlDbType.NVarChar).Value = txt\_NhanVien\_DiaChi.Text.Trim();  cmd.Parameters.Add("@SDT", SqlDbType.NVarChar).Value = txt\_NhanVien\_SDT.Text.Trim();  cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = txt\_NhanVien\_Email.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận sửa?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Sửa thành công");  LoadNhanVien();  }  else  {  MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Xem nhân viên**

|  |
| --- |
| -- 3. Xem chi tiết thông tin Nhân Viên  CREATE OR ALTER VIEW view\_CTThongTinNV AS  SELECT  NV.MaNhanVien AS MaNhanVien,  NV.HoTen AS HoTen,  NV.CCCD AS CCCD,  NV.SDT AS SDT,  NV.DiaChi AS DiaChi,  NV.Email AS Email  FROM NhanVien NV  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void LoadNhanVien()  {  string sql = "SELECT \* FROM view\_CTThongTinNV";  dgv\_NhanVien.DataSource = sqlConnect.ExecuteQuery(sql);  }  public void XemNhanVien()  {  try  {  sqlConnect.OpenConnection();  LoadNhanVien();  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

* 1. **Tìm kiếm nhân viên**
* **Theo “Mã nhân viên”**

|  |
| --- |
| -- func13 Tìm nhân viên theo mã  CREATE OR ALTER FUNCTION func\_TimNhanVienTheoMa  (@MaNhanVien NVARCHAR(10))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM NhanVien  WHERE MaNhanVien = @MaNhanVien  )  GO |

* **Theo “Tên nhân viên”**

|  |
| --- |
| -- fun14 Tìm nhân viên theo tên  CREATE OR ALTER FUNCTION func\_TimNhanVienTheoTen  (@HoTen NVARCHAR(50))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM NhanVien  WHERE HoTen LIKE '%' + @HoTen + '%'  )  GO |

* **Theo “Số điện thoại”**

|  |
| --- |
| -- fun15 Tìm nhân viên theo SĐT  CREATE OR ALTER FUNCTION func\_TimNhanVienTheoSDT  (@SDT NVARCHAR(10))  RETURNS TABLE  AS  RETURN (  SELECT \*  FROM NhanVien  WHERE SDT = @SDT  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void TimKiemNhanVien()  {  string tuKhoa = txt\_NhanVien\_TimKiem.Text.Trim();  if (string.IsNullOrEmpty(tuKhoa))  {  MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (rbt\_NhanVien\_MaNhanVien.Checked)  {  string query = "SELECT \* FROM func\_TimNhanVienTheoMa(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_NhanVien.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_NhanVien\_TenNhanVien.Checked)  {  string query = "SELECT \* FROM func\_TimNhanVienTheoTen(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_NhanVien.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else if (rbt\_NhanVien\_SDT.Checked)  {  string query = "SELECT \* FROM func\_TimNhanVienTheoSDT(@TuKhoa)";  SqlCommand cmd = new SqlCommand(query, sqlConnect.con);  cmd.Parameters.Add(new SqlParameter("@TuKhoa", tuKhoa));  try  {  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dgv\_NhanVien.DataSource = dt;  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  MessageBox.Show("Không tìm thấy!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

* 1. **Xóa Nhân viên**

|  |
| --- |
| -- proc14 Xóa nhân viên  CREATE OR ALTER PROCEDURE proc\_XoaNhanVien  @MaNhanVien NVARCHAR(10)  AS  BEGIN  -- Kiểm tra xem nhân viên có tồn tại không  IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNhanVien = @MaNhanVien)  BEGIN  RAISERROR(N'Nhân Viên không tồn tại!', 16, 1);  RETURN;  END  -- Xóa Nhân viên từ bảng NhanVien  DELETE FROM NhanVien WHERE MaNhanVien = @MaNhanVien;  END  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void XoaNhanVien()  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_XoaNhanVien", sqlConnect.con);  cmd.Parameters.Add("@MaNhanVien", SqlDbType.NVarChar).Value = txt\_NhanVien\_MaNhanVien.Text.Trim();  cmd.CommandType = CommandType.StoredProcedure;  DialogResult result = MessageBox.Show("Xác nhận xóa", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Xóa nhân viên thành công");  LoadNhanVien();  }  }  else  {  MessageBox.Show("Xóa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

1. **Quản lý thống kê**

### **Thống kê sách**

|  |
| --- |
| -- proc15 Thống kê sách được mượn nhiều nhất theo theo lượng giảm dần  CREATE OR ALTER PROCEDURE proc\_ThongKeSachDuocMuonNhieuNhat  AS  BEGIN  SELECT  DS.MaSach,  DS.TenSach,  COUNT(CT.MaCuon) AS SoLuongMuon  FROM DauSach DS  INNER JOIN CuonSach CS ON DS.MaSach = CS.MaSach  INNER JOIN CT\_MuonTra CT ON CS.MaCuon = CT.MaCuon  GROUP BY DS.MaSach, DS.TenSach  ORDER BY SoLuongMuon DESC;  END;  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeSach\_SachMuonNhieuNhat\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachDuocMuonNhieuNhat", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

|  |
| --- |
| -- proc16 Thống kê sách theo thể loại  CREATE PROCEDURE proc\_ThongKeSachTheoTheLoai  AS  BEGIN  SELECT  TL.MaTheLoai AS MaTheLoai,  TL.TenTheLoai AS TenTheLoai,  COUNT(DS.MaSach) AS SoLuongSach  FROM TheLoai TL  INNER JOIN DauSach DS ON TL.MaTheLoai = DS.MaTheLoai  GROUP BY TL.MaTheLoai, TL.TenTheLoai  END;  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeSach\_TheLoai\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachTheoTheLoai", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

|  |
| --- |
| -- proc17 Thống kê sách theo tên tác giả  CREATE OR ALTER PROCEDURE proc\_ThongKeSachTheoTacGia  AS  BEGIN  SELECT  TG.MaTacGia AS MaTacGia,  TG.TenTacGia AS TenTacGia,  COUNT(DS.MaSach) AS SoLuongSach  FROM TacGia TG  INNER JOIN DauSach DS ON TG.MaTacGia = DS.MaTacGia  GROUP BY TG.MaTacGia, TG.TenTacGia  END;  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeSach\_TacGia\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachTheoTacGia", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* 1. **Thống kê độc giả**

|  |
| --- |
| - func18 Thống kê độc giả trả sách quá hạn  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaTraSachTreHan()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.SoNgayQuaHan  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE CTMT.SoNgayQuaHan > 0  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_TraTreHan\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaTraSachTreHan()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

|  |
| --- |
| -- func18 Thống kê độc giả làm hư sách  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaLamHuSach()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.TinhTrang,  CTMT.GhiChu  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE (CTMT.TinhTrang = N'Hư')  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_Hu\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaLamHuSach()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

|  |
| --- |
| -- func17 Thống kê độc giả làm mất sách  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaLamMatSach()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.TinhTrang,  CTMT.GhiChu  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE ( CTMT.TinhTrang = N'Mất')  )  GO |

* **Thực thi trên C#**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_Mat\_CheckedChanged(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaLamMatSach()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

Ứng với hai nhóm người dung (Nhân viên và quản lý) là các role riêng :

* **Đối với Role Staff (dành cho nhân viên):**

|  |
| --- |
| -- PHÂN QUYỀN  CREATE ROLE r\_staff;  GRANT SELECT ON DauSach TO r\_staff  GRANT SELECT ON CuonSach TO r\_staff  GRANT SELECT ON NXB TO r\_staff  GRANT SELECT ON TheLoai TO r\_staff  GRANT SELECT ON TacGia TO r\_staff  GRANT SELECT, INSERT, DELETE, UPDATE ON DocGia TO r\_staff  GRANT SELECT, INSERT, DELETE, UPDATE ON PhieuMuonTra TO r\_staff  GRANT SELECT, INSERT, DELETE, UPDATE ON CT\_MuonTra TO r\_staff  GRANT SELECT TO r\_staff  GRANT EXEC ON proc\_SuaDocGia TO r\_staff  GRANT EXEC ON proc\_SuaPhieuMuon TO r\_staff  GRANT EXEC ON proc\_SuaPhieuTra TO r\_staff  GRANT EXEC ON proc\_ThemDocGia TO r\_staff  GRANT EXEC ON proc\_ThemPhieuMuon TO r\_staff  GRANT EXEC ON proc\_ThemPhieuTra TO r\_staff  GRANT EXEC ON proc\_XoaDocGia TO r\_staff  GRANT EXEC ON proc\_XoaPhieuTra TO r\_staff  GRANT EXEC ON proc\_ThayDoiMatKhau TO r\_staff  DENY EXEC ON proc\_ThongKeSachDuocMuonNhieuNhat to r\_staff  DENY EXEC ON proc\_ThongKeSachTheoTacGia to r\_staff  DENY EXEC ON proc\_ThongKeSachTheoTheLoai to r\_staff  DENY EXEC ON proc\_ThemSach to r\_staff  DENY EXEC ON proc\_SuaSach to r\_staff  DENY EXEC ON proc\_XoaSach to r\_staff  DENY EXEC ON proc\_ThemNhanVien to r\_staff  DENY EXEC ON proc\_SuaNhanVien to r\_staff  DENY EXEC ON proc\_XoaNhanVien to r\_staff |

* **Đối với Role Manager ta sử dụng server role sysadmin.**
* **Trigger tạo tài khoản**

|  |
| --- |
| -- Trigger tạo tài khoản  CREATE OR ALTER TRIGGER trg\_TaoTaiKhoan  ON DangNhap  AFTER INSERT  AS  BEGIN  DECLARE @sql VARCHAR(1000);  DECLARE @TenDangNhap VARCHAR(20);  DECLARE @MatKhau VARCHAR(20);  SELECT @TenDangNhap = LTRIM(RTRIM(i.TenDangNhap)), @MatKhau = LTRIM(RTRIM(i.MatKhau)) FROM inserted i;  SET @sql = 'CREATE LOGIN [' + @TenDangNhap +'] WITH PASSWORD = ''' + @MatKhau  + ''' , DEFAULT\_DATABASE=[Nhom8\_QLTV], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF';  EXEC (@sql);  SET @sql = 'CREATE USER ' + @TenDangNhap +' FOR LOGIN '+ @TenDangNhap;  EXEC (@sql);  SET @sql = 'USE master; ' +  'GRANT ALTER ANY LOGIN TO ' + QUOTENAME(@TenDangNhap);  EXEC (@sql);  SET @sql ='USE Nhom8\_QLTV ';  EXEC (@sql);  IF EXISTS (SELECT MaNhanVien FROM Nha  Vien WHERE @TenDangNhap = 'NV01')  BEGIN  SET @sql = 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @TenDangNhap;  END;  IF EXISTS (SELECT MaNhanVien FROM NhanVien WHERE MaNhanVien = @TenDangNhap AND @TenDangNhap <> 'NV01')  BEGIN  SET @sql = 'ALTER ROLE r\_staff ADD MEMBER ' + @TenDangNhap;  END;  EXEC (@sql);  END;  GO |

**- Trigger Xóa tài khoản**

|  |
| --- |
| -- Trigger xóa tài khoản  CREATE PROC proc\_XoaTaiKhoan  @TenDangNhap NVARCHAR(10)  AS  DECLARE @sql NVARCHAR(2000);  DECLARE @sessionID INT;  SELECT @sessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @TenDangNhap;  IF @sessionID IS NOT NULL  BEGIN  SET @sql = 'KILL ' + CONVERT(NVARCHAR(20), @sessionID);  EXEC (@sql);  END;  BEGIN  BEGIN TRY  SET @sql = 'DROP USER ' + @TenDangNhap;  EXEC (@sql);  SET @sql = 'DROP LOGIN ' + @TenDangNhap;  EXEC (@sql);  DELETE FROM DangNhap WHERE TenDangNhap = @TenDangNhap;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX);  SELECT @err = ERROR\_MESSAGE();  RAISERROR(@err, 16, 1);  END CATCH;  END;  GO |

**- Trigger Thay đổi mật khẩu**

|  |
| --- |
| -- Trigger thay đổi mật khẩu  CREATE OR ALTER PROC proc\_ThayDoiMatKhau  @TenDangNhap VARCHAR(10),  @MatKhau VARCHAR(20)  AS  BEGIN  DECLARE @sql NVARCHAR(MAX);  DECLARE @user NVARCHAR(10);  DECLARE @newPassword NVARCHAR(20);  SET @user = LTRIM(RTRIM(@TenDangNhap));  SET @newPassword = LTRIM(RTRIM(@MatKhau));  BEGIN TRANSACTION;  BEGIN TRY  SET @sql = 'ALTER LOGIN [' + @user + '] WITH PASSWORD = ''' + @newPassword + '''';  EXEC(@sql);  UPDATE DangNhap SET MatKhau = @newPassword WHERE TenDangNhap = @user;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX);  SET @err ='Không cập nhập được mật khẩu mới!';  RAISERROR(@err, 16, 1);  ROLLBACK TRANSACTION;  THROW;  END CATCH;  COMMIT TRANSACTION;  END;  GO |

**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

Môi trường cài đặt và công nghệ sử dụng

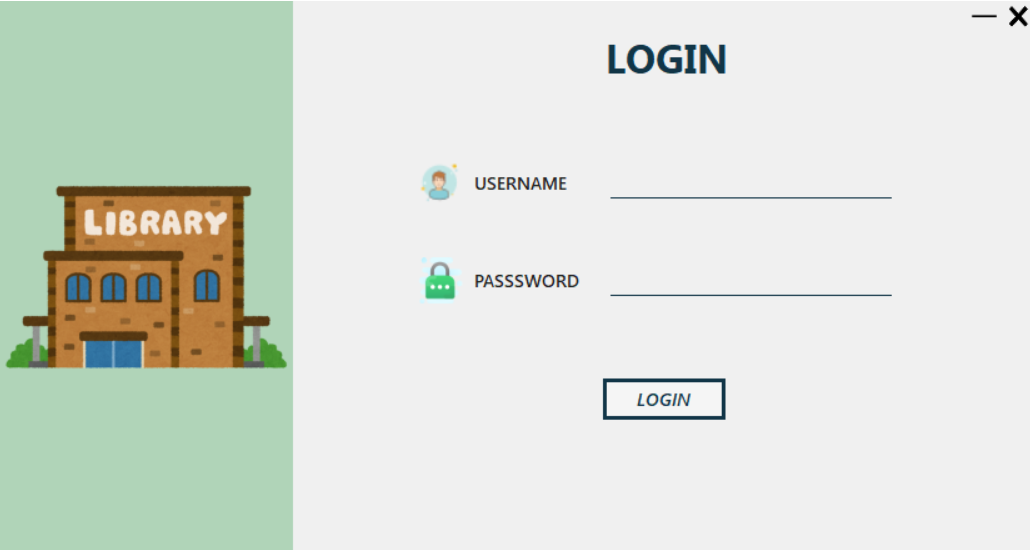
- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng

ASP.NET Web Application (.NET Framework) C#.

Giao diện và chức năng của chương trình:

**Trang login**



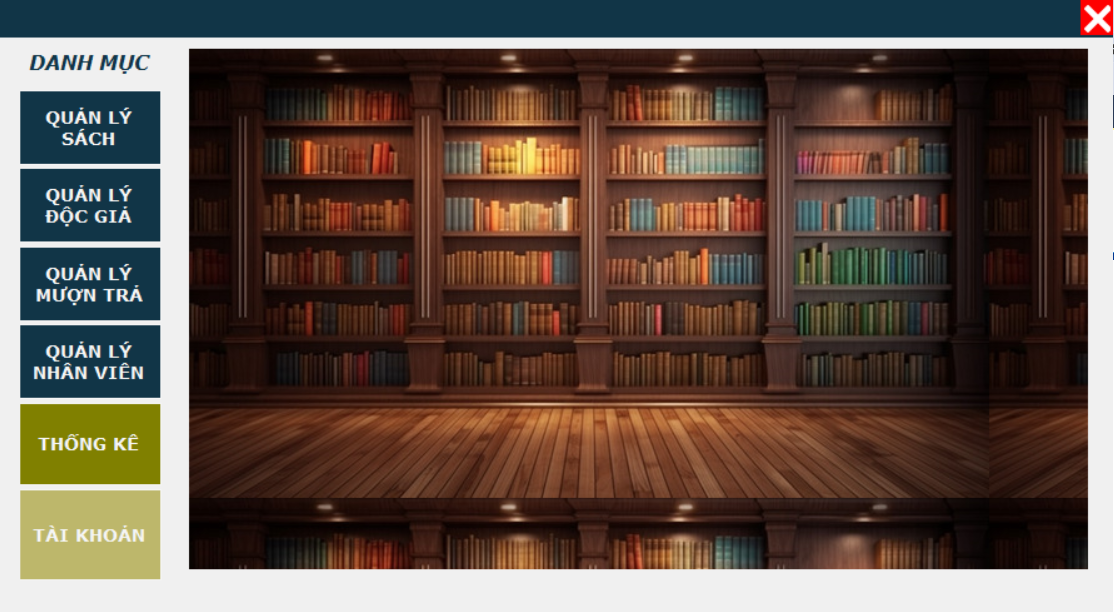
* **Bảng Login – Trigger thêm dữ liệu sau khi insert nhân viên**

|  |
| --- |
| CREATE TABLE DangNhap  (  TenDangNhap NVARCHAR(10) PRIMARY KEY REFERENCES NhanVien(MaNhanVien),  MatKhau NVARCHAR(50) NOT NULL,  CONSTRAINT FK\_DangNhap\_NhanVien FOREIGN KEY (TenDangNhap) REFERENCES NhanVien(MaNhanVien) ON UPDATE CASCADE ON DELETE CASCADE  );  GO  -- Trigger thêm bảng vào đăng nhập  CREATE OR ALTER TRIGGER trg\_TaoTaiKhoanNhanVien  ON NhanVien  AFTER INSERT  AS  BEGIN  DECLARE @TenDangNhap NVARCHAR(10);  DECLARE @MatKhau NVARCHAR(10);  SELECT @TenDangNhap = i.MaNhanVien, @MatKhau = i.MaNhanVien  FROM inserted i;  INSERT INTO DangNhap VALUES (@TenDangNhap, @MatKhau);  END;  GO |

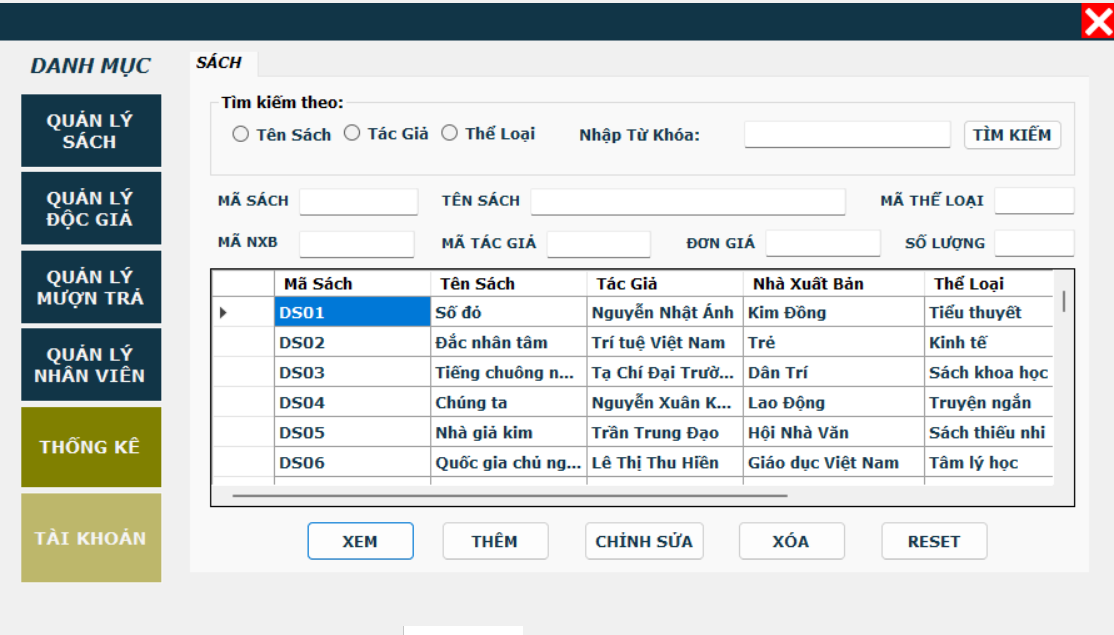
* **Thực thi C#:**

|  |
| --- |
| public partial class Login : Form  {  public Login()  {  InitializeComponent();  }  SQLConnect.SqlConnect sqlConnect = new SQLConnect.SqlConnect();  private void btn\_login\_Click(object sender, EventArgs e)  {  string username = this.txt\_account.Text;  string password = this.txt\_password.Text;  sqlConnect.OpenConnection();  DataTable dt = new DataTable();  string sql = "select \* from DangNhap where TenDangNhap = @TenDangNhap";  SqlCommand cmd = new SqlCommand(sql, sqlConnect.con);  cmd.CommandType = CommandType.Text;  SqlParameter \_username = new SqlParameter("@TenDangNhap", SqlDbType.VarChar);  \_username.Value = username;  cmd.Parameters.Add(\_username);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dt);  if (dt.Rows.Count > 0)  {  if (dt.Rows[0]["MatKhau"].ToString() == password)  {  string strConnect = $"Data Source=.;Initial Catalog=Nhom8\_QLTV;User ID={username};Password={password}";  SQLConnect.SqlConnect.connectionString= strConnect ;  this.Hide();  Form1 form1 = new Form1();  form1.ShowDialog();  }  else  {  MessageBox.Show("Password không đúng!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  else  {  MessageBox.Show("UserName không tồn tại!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  sqlConnect.CloseConnection();  }    private void btn\_Thoat\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Bạn có muốn thoát?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)  {  Application.Exit();  }  }  } |

**Trang home**



**Trang quản lý sách**



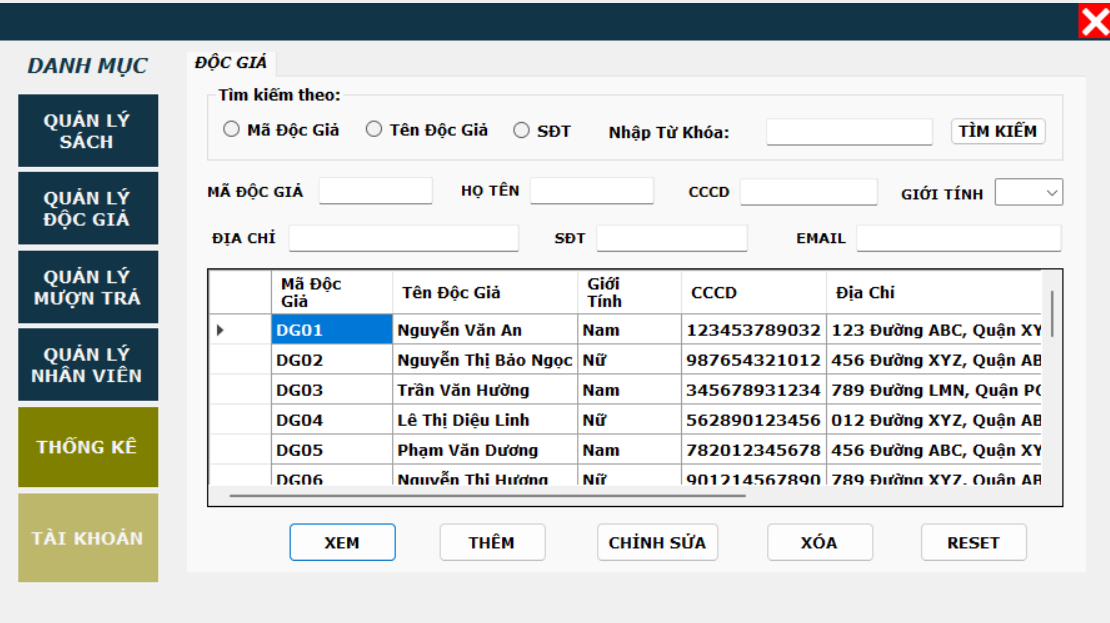
* **Danh mục Sách**

|  |
| --- |
| CREATE OR ALTER VIEW view\_CTThongTinSach AS  SELECT  DS.MaSach AS MaSach,  DS.TenSach AS TenSach,  TG.TenTacGia AS TenTacGia,  NXB.TenNXB AS TenNXB,  TL.TenTheLoai AS TenTheLoai,  DS.Gia AS Gia,  DS.SoLuong AS SoLuong,  TG.MaTacGia AS MaTacGia,  NXB.MaNXB AS MaNXB,  TL.MaTheLoai AS MaTheLoai  FROM DauSach DS  JOIN TacGia TG ON DS.MaTacGia = TG.MaTacGia  JOIN NXB ON DS.MaNXB = NXB.MaNXB  JOIN TheLoai TL ON DS.MaTheLoai = TL.MaTheLoai  GO |

* **Thực thi trên C#:**

|  |
| --- |
| public void XemSach()  {  try  {  sqlConnect.OpenConnection();  string sql = "SELECT \* FROM view\_CTThongTinSach";  dgv\_Sach.DataSource = sqlConnect.ExecuteQuery(sql);  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  } |

**Trang quản lý độc giả**



* **Danh mục Độc Giả**

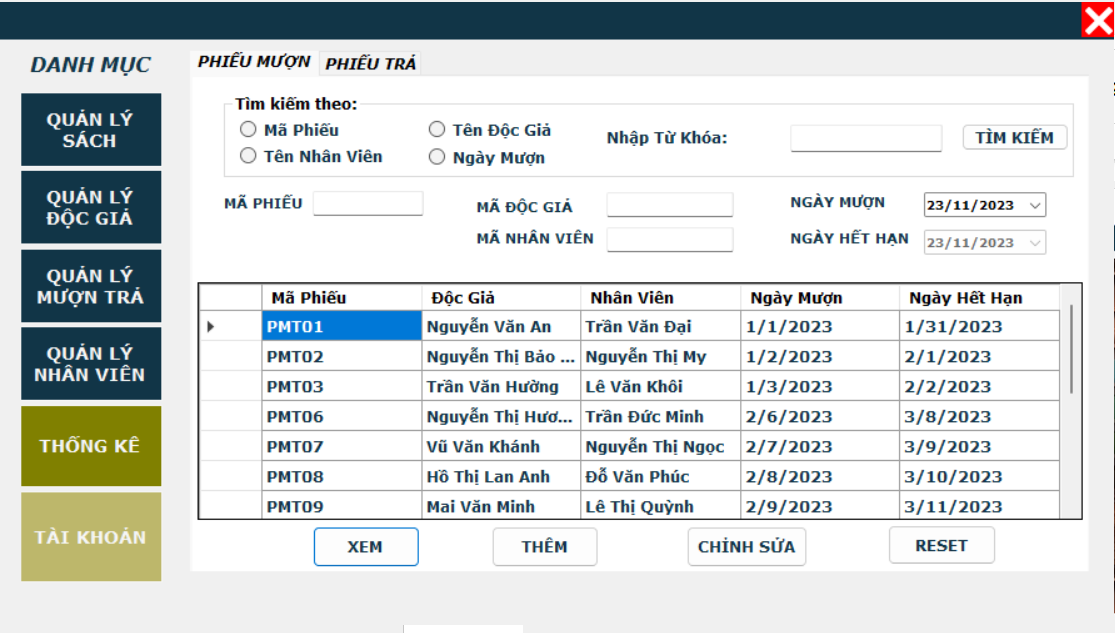
|  |
| --- |
| CREATE OR ALTER VIEW view\_CTThongTinDocGia AS  SELECT  DG.MaDocGia AS MaDocGia,  DG.TenDocGia AS TenDocGia,  DG.GioiTinh AS GioiTinh,  DG.CCCD AS CCCD,  DG.DiaChi AS DiaChi,  DG.SDT AS SDT,  DG.Email AS Email  FROM DocGia DG  GO |

* **Thực thi trên C#:**

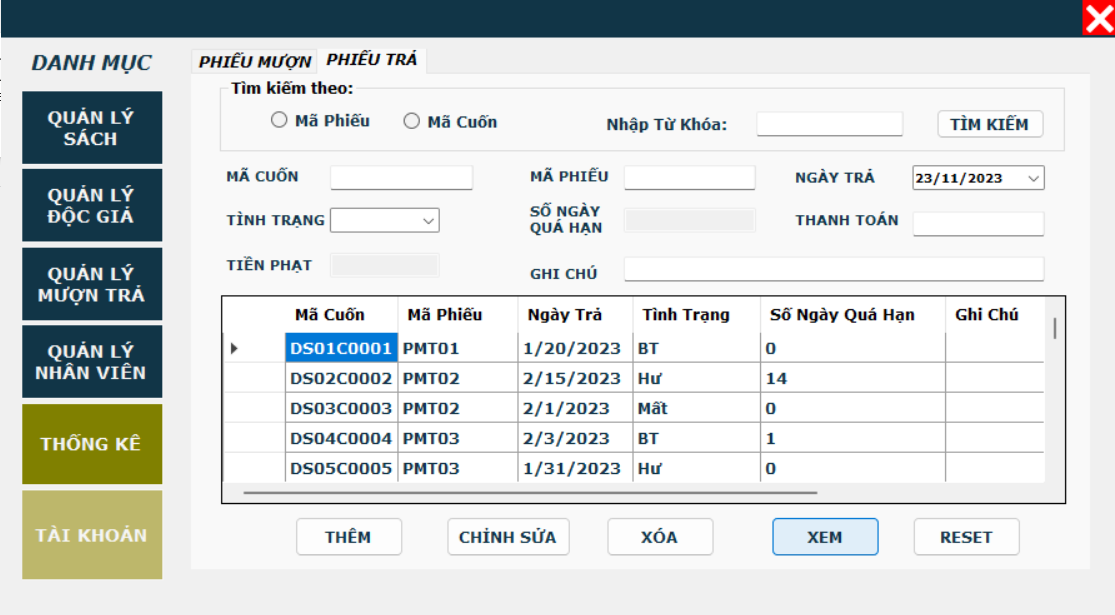
|  |
| --- |
| public void XemDocGia()  {  try  {  sqlConnect.OpenConnection();  string sql = "SELECT \* FROM view\_CTThongTinDocGia";  dgv\_DocGia.DataSource = sqlConnect.ExecuteQuery(sql);  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**Trang quản lý phiếu mượn trả**

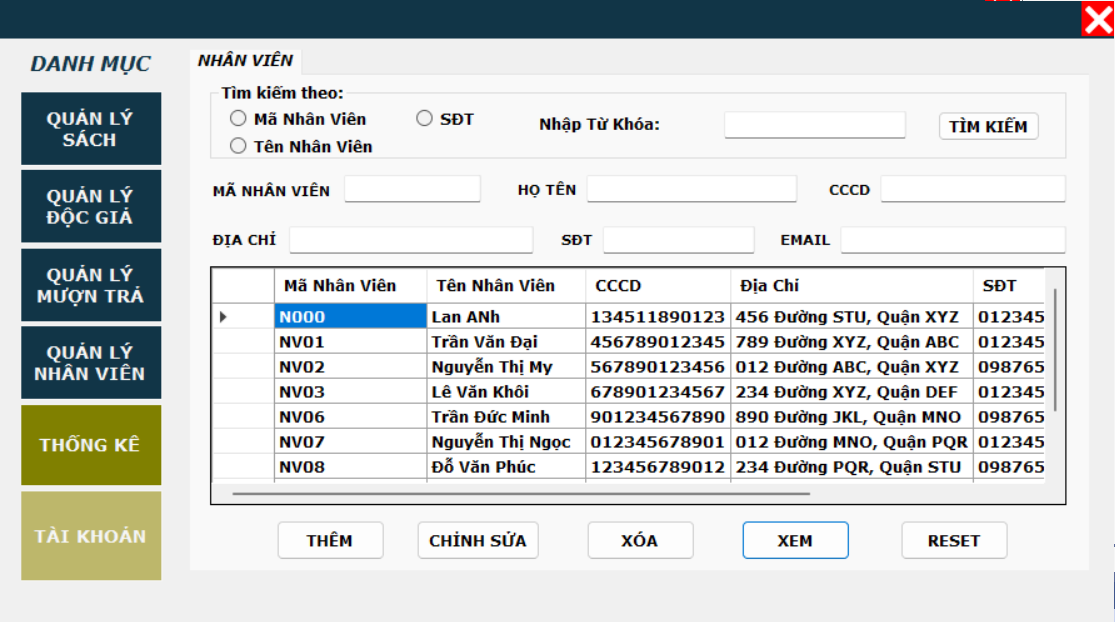
* 1. **Quản lý phiếu mượn**



### **Quản lý phiếu trả (Chi tiết phiếu mượn trả)**



**Quản lý nhân viên**



* **Code Sql:**

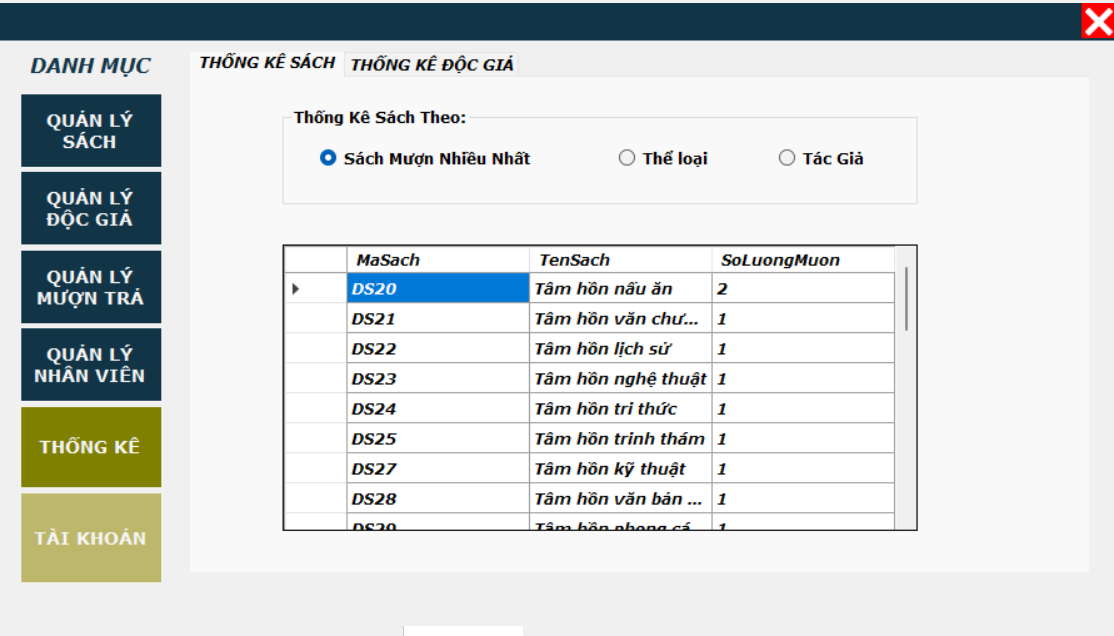
|  |
| --- |
| -- 3. Xem chi tiết thông tin Nhân Viên  CREATE OR ALTER VIEW view\_CTThongTinNV AS  SELECT  NV.MaNhanVien AS MaNhanVien,  NV.HoTen AS HoTen,  NV.CCCD AS CCCD,  NV.SDT AS SDT,  NV.DiaChi AS DiaChi,  NV.Email AS Email  FROM NhanVien NV  GO |

* **Thực thi trên C#**

|  |
| --- |
| public void LoadNhanVien()  {  string sql = "SELECT \* FROM view\_CTThongTinNV";  dgv\_NhanVien.DataSource = sqlConnect.ExecuteQuery(sql);  }  public void XemNhanVien()  {  try  {  sqlConnect.OpenConnection();  LoadNhanVien();  sqlConnect.CloseConnection();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**Trang quản lý thống kê**

* 1. **Thống kê sách**
     1. **Thống kê sách được mượn nhiều nhất**



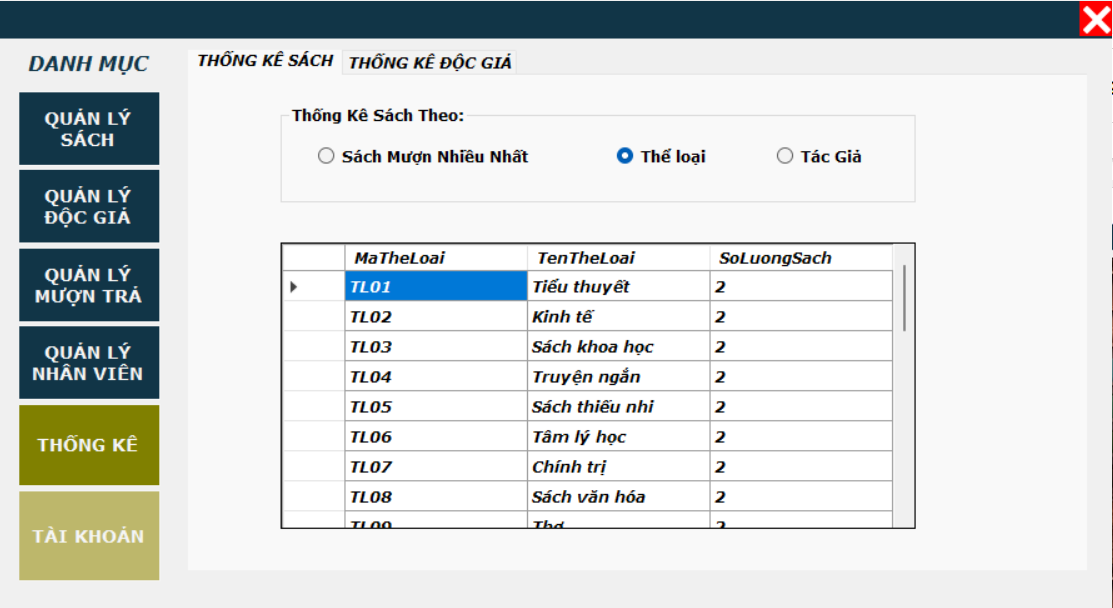
* **Code Sql:**

|  |
| --- |
| -- proc15 Thống kê sách được mượn nhiều nhất theo theo lượng giảm dần  CREATE OR ALTER PROCEDURE proc\_ThongKeSachDuocMuonNhieuNhat  AS  BEGIN  SELECT  DS.MaSach,  DS.TenSach,  COUNT(CT.MaCuon) AS SoLuongMuon  FROM DauSach DS  INNER JOIN CuonSach CS ON DS.MaSach = CS.MaSach  INNER JOIN CT\_MuonTra CT ON CS.MaCuon = CT.MaCuon  GROUP BY DS.MaSach, DS.TenSach  ORDER BY SoLuongMuon DESC;  END;  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeSach\_SachMuonNhieuNhat\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachDuocMuonNhieuNhat", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* + 1. **Thống kê sách theo thể loại**



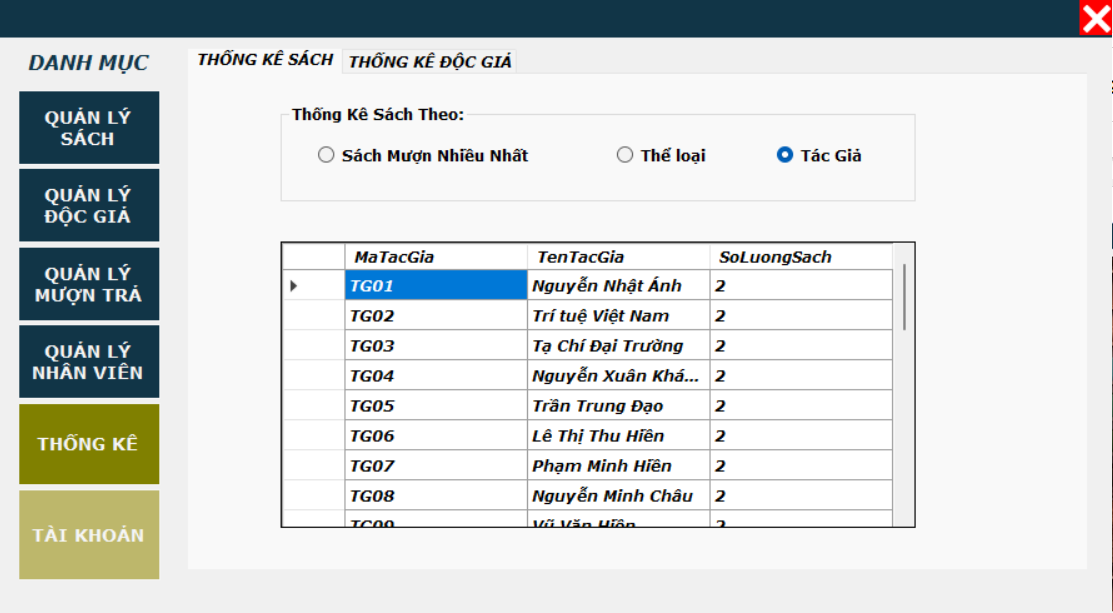
* **Code Sql:**

|  |
| --- |
| -- proc16 Thống kê sách theo thể loại  CREATE PROCEDURE proc\_ThongKeSachTheoTheLoai  AS  BEGIN  SELECT  TL.MaTheLoai AS MaTheLoai,  TL.TenTheLoai AS TenTheLoai,  COUNT(DS.MaSach) AS SoLuongSach  FROM TheLoai TL  INNER JOIN DauSach DS ON TL.MaTheLoai = DS.MaTheLoai  GROUP BY TL.MaTheLoai, TL.TenTheLoai  END;  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeSach\_TheLoai\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachTheoTheLoai", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* + 1. **Thống kê sách theo tác giả**



* **Code Sql:**

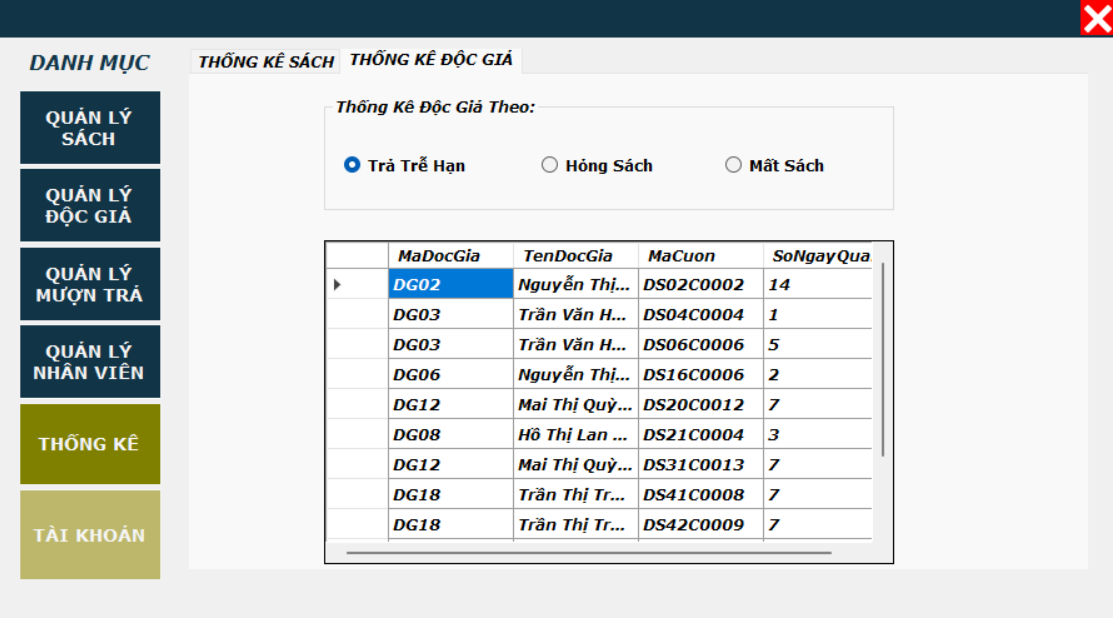
|  |
| --- |
| -- proc17 Thống kê sách theo tên tác giả  CREATE OR ALTER PROCEDURE proc\_ThongKeSachTheoTacGia  AS  BEGIN  SELECT  TG.MaTacGia AS MaTacGia,  TG.TenTacGia AS TenTacGia,  COUNT(DS.MaSach) AS SoLuongSach  FROM TacGia TG  INNER JOIN DauSach DS ON TG.MaTacGia = DS.MaTacGia  GROUP BY TG.MaTacGia, TG.TenTacGia  END;  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeSach\_TacGia\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThongKeSachTheoTacGia", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeSach.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### **Thống kê độc giả**

#### **Thống kê độc giả trả trễ hạn**



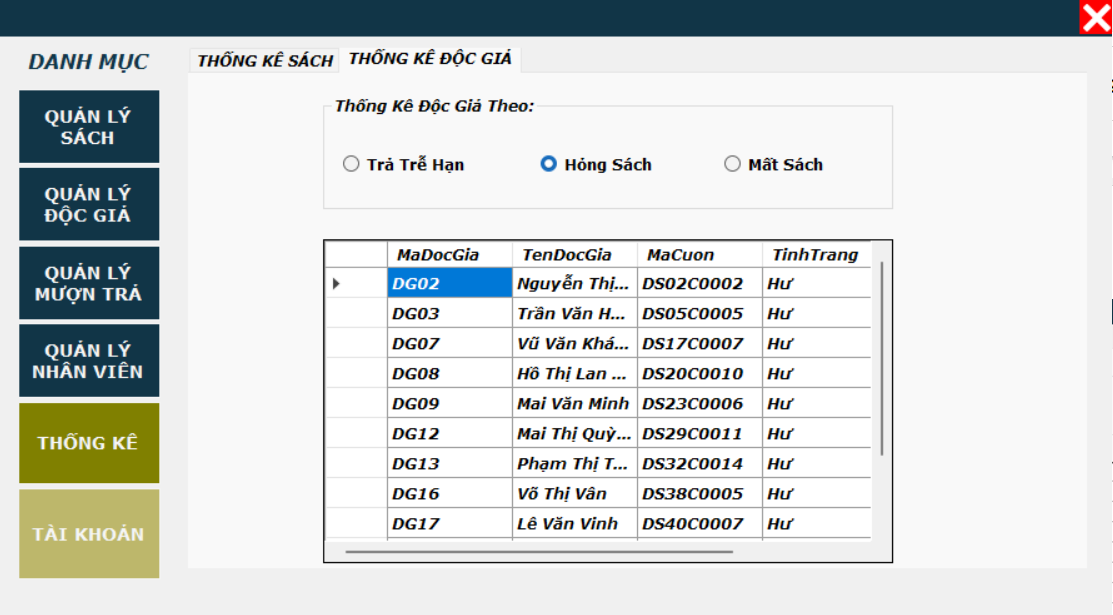
* **Code Sql:**

|  |
| --- |
| - func18 Thống kê độc giả trả sách quá hạn  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaTraSachTreHan()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.SoNgayQuaHan  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE CTMT.SoNgayQuaHan > 0  )  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_TraTreHan\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaTraSachTreHan()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### **Thống kê độc giả làm hư sách**



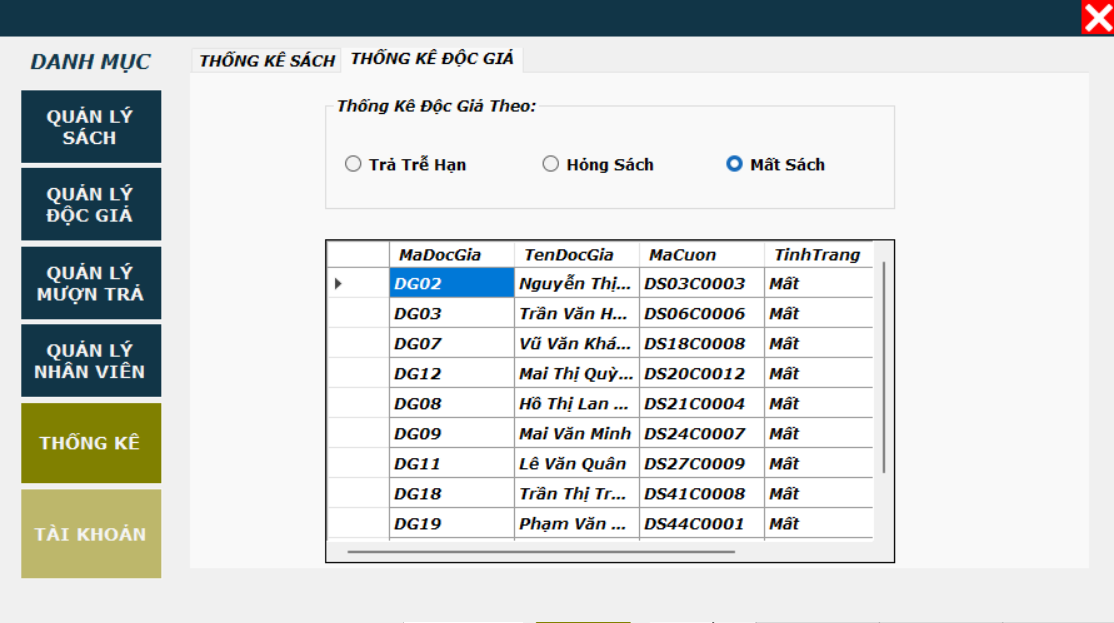
* **Code Sql:**

|  |
| --- |
| -- func18 Thống kê độc giả làm hư sách  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaLamHuSach()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.TinhTrang,  CTMT.GhiChu  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE (CTMT.TinhTrang = N'Hư')  )  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_Hu\_CheckedChanged\_1(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaLamHuSach()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### **Thống kê độc giả làm mất sách**



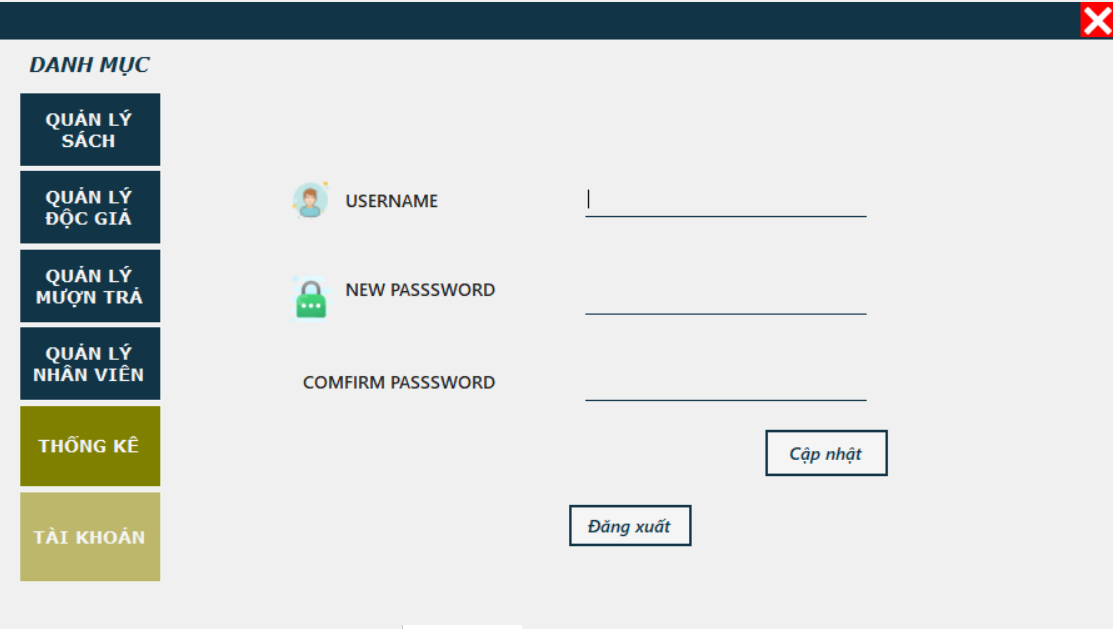
* **Code Sql:**

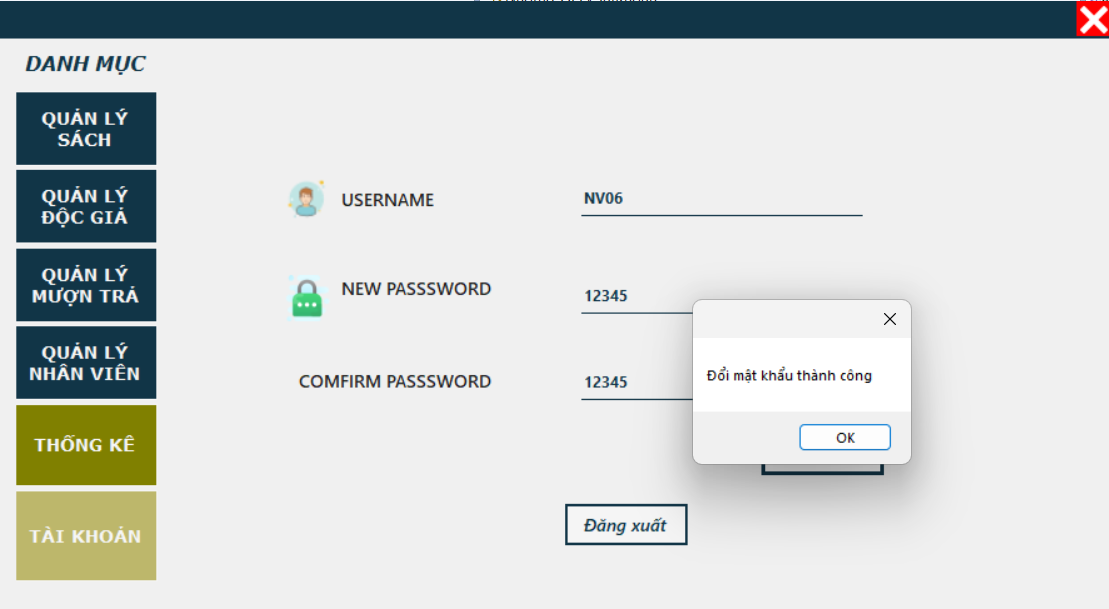
|  |
| --- |
| -- func17 Thống kê độc giả làm mất sách  CREATE OR ALTER FUNCTION func\_ThongKeDocGiaLamMatSach()  RETURNS TABLE  AS  RETURN (  SELECT  DG.MaDocGia,  DG.TenDocGia,  CTMT.MaCuon,  CTMT.TinhTrang,  CTMT.GhiChu  FROM DocGia DG  INNER JOIN PhieuMuonTra PMT ON DG.MaDocGia = PMT.MaDocGia  INNER JOIN CT\_MuonTra CTMT ON PMT.MaPhieu = CTMT.MaPhieu  WHERE ( CTMT.TinhTrang = N'Mất')  )  GO |

* **Thực thi trên C#:**

|  |
| --- |
| private void rbt\_ThongKeDocGia\_Mat\_CheckedChanged(object sender, EventArgs e)  {  try  {  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM func\_ThongKeDocGiaLamMatSach()", sqlConnect.con);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  dgv\_ThongKeDocGia.DataSource = dt;  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

1. **Trang tài khoản (Cho phép đổi password)**





* **Code sql:**

|  |
| --- |
| -- Trigger thay đổi mật khẩu  CREATE OR ALTER PROC proc\_ThayDoiMatKhau  @TenDangNhap VARCHAR(10),  @MatKhau VARCHAR(20)  AS  BEGIN  DECLARE @sql NVARCHAR(MAX);  DECLARE @user NVARCHAR(10);  DECLARE @newPassword NVARCHAR(20);  SET @user = LTRIM(RTRIM(@TenDangNhap));  SET @newPassword = LTRIM(RTRIM(@MatKhau));  BEGIN TRANSACTION;  BEGIN TRY  SET @sql = 'ALTER LOGIN [' + @user + '] WITH PASSWORD = ''' + @newPassword + '''';  EXEC(@sql);  UPDATE DangNhap SET MatKhau = @newPassword WHERE TenDangNhap = @user;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX);  SET @err ='Không cập nhập được mật khẩu mới!';  RAISERROR(@err, 16, 1);  ROLLBACK TRANSACTION;  THROW;  END CATCH;  COMMIT TRANSACTION;  END;  GO |

* **Thực hiện C#:**

|  |
| --- |
| public partial class TaiKhoan : UserControl  {  public TaiKhoan()  {  InitializeComponent();  }  SQLConnect.SqlConnect sqlConnect = new SQLConnect.SqlConnect();  private void btn\_CapNhat\_Click(object sender, EventArgs e)  {  try  {  string userName = txt\_userName.Text;  string password = txt\_password.Text;  string comfirmpassword = txt\_comfirmpassword.Text;  if (string.IsNullOrEmpty(userName) || string.IsNullOrEmpty(password) || string.IsNullOrEmpty(comfirmpassword))  {  MessageBox.Show("Vui lòng điền đầy đủ thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  if (password != comfirmpassword)  {  MessageBox.Show("Mật khẩu không trùng khớp, vui lòng nhập lại!.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  sqlConnect.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ThayDoiMatKhau", sqlConnect.con);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@TenDangNhap", SqlDbType.NVarChar).Value = txt\_userName.Text.Trim();  cmd.Parameters.Add("@MatKhau", SqlDbType.NVarChar).Value = txt\_comfirmpassword.Text.Trim();  DialogResult result = MessageBox.Show("Xác nhận đổi mật khẩu?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Đổi mật khẩu thành công");  }  else  {  MessageBox.Show("Thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  sqlConnect.CloseConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |