**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



**TÊN ĐỀ TÀI: Quản Lý Thư Viện**

Môn học: Hệ quản trị cơ sở dữ liệu

Giảng viên hướng dẫn: TS. Nguyễn Thành Sơn

Nhóm 8:

Trần Trung Phát 20110536

Lê Quốc Văn 21110862

Trần Khải Hoàn 21110827

Đoàn Nguyễn Nam Trung 21110333

Hồ Chí Minh, tháng 5 năm 2024

**NHẬN XÉT CỦA GIẢNG VIÊN**

Tp. Hồ Chí Minh, ngày…. tháng…. năm 2024

Giảng viên chấm điểm

**LỜI CẢM ƠN**

Trên thực tế, không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian học tập tại trường đến nay, nhóm em đã nhận được sự quan tâm, giúp đỡ của Thầy Cô, gia đình và bạn bè.

Và đặc biệt, trong học kỳ này, em gửi lời cảm ơn tới thầy TS. Nguyễn Thành Sơn đã hướng dẫn, góp ý, thảo luận để báo cáo môn Hệ Quản Trị Cơ Sở Dữ Liệu của chúng em được hoàn thiện. Trong thời gian được học tập và thực hành dưới sự hướng dẫn của thầy, chúng em không những tiếp thu được rất nhiều kiến thức bổ ích, mà còn tích lũy được một số kinh nghiệm.

Mặc dù đã rất cố gắng hoàn thiện bài báo cáo với tất cả sự nỗ lực, tuy nhiên đề tài “Xây dựng chương trình quản lý thư viện” sẽ khó tránh khỏi những thiếu sót trong quá trình xây dựng. Chúng em rất mong nhận được sự quan tâm, thông cảm và những đóng góp quý báu của các thầy để báo cáo môn học Hệ Quản Trị Cơ Sở Dữ Liệu của nhóm chúng em ngày càng hoàn thiện hơn.

Một lần nữa, chúng em xin chân thành cảm ơn thầy và luôn mong nhận được sự đóng góp của thầy và các thầy cô khác.

Sau cùng, chúng em xin kính chúc thầy cô trong khoa Công Nghệ Thông Tin dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao cả của mình truyền đạt kiến thức cho thế hệ mai sau.

Trân Trọng!

**Mục lục**

[**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 8**](#_e2e4git1f1bm)

[1.1. Đặc tả đề tài 8](#_6i9fosjbfg19)

[1.1.1 Mô tả bài toán 8](#_msugu81lzchh)

[1.1.2. Nghiệp vụ của bài toán 9](#_xicpf9dyjwcl)

[1.1.3. Mô tả chức năng của bài toán 11](#_n7176d6dqdol)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 13**](#_rp0ttvymk36a)

[2.1. Thiết kế cơ sở dữ liệu mức quan niệm 13](#_64ldgxaha8lx)

[2.2. Thiết kế cơ sở dữ liệu mức logic 14](#_yrac1etxqy8v)

[2.3. Các ràng buộc cần có 14](#_jp611fkigf9p)

[2.4. Cài đặt các CSDL và các ràng buộc 15](#_o9640jy62xcg)

[2.5. Các view 19](#_qpjsorfxh0ap)

[2.5.1. View danh sách tài khoản 19](#_ur3zbfwiv02e)

[2.5.2. View danh sách lịch làm việc 19](#_zb7ugcskgqrw)

[2.5.3. View danh sách thủ thư 19](#_gri8viuk1u7z)

[2.5.4. View danh sách phiếu phạt 19](#_yechh2kir17y)

[2.5.5. View danh sách phiếu mượn sách 20](#_fq8cvkbxduqp)

[2.5.6. View danh sách sách thư viện 20](#_kwac9jykz9f1)

[2.5.7. View danh sách tác giả 20](#_p11u66evno8h)

[2.5.8. View danh sách nhà xuất bản 21](#_anbg6uf44j6k)

[2.5.9. View danh sách thể loại 21](#_gmul4ii8s1cf)

[2.6. Các Trigger 21](#_1dxdsjc6buft)

[2.6.1. Trigger ngăn bị trùng số điện thoại 21](#_yytg9tdbua5m)

[2.6.2. Trigger giới hạn giờ làm việc trong một tuần của thủ thư 22](#_vgrsyjlq9gdk)

[2.6.3. Trigger giới hạn trùng lặp cùng ngày cùng ca làm của một thủ thư 23](#_xki562lhh9s1)

[2.6.4. Trigger cập nhật số lượng sách khi sau khi tạo phiếu mượn 23](#_dj5a9bergk2u)

[2.6.5. Trigger giới hạn trùng lặp tên thể loại 24](#_8r60wh6gcz0s)

[2.6.6. Trigger giới hạn không cho ngày trả phạt nhỏ hơn ngày trả sách 25](#_9wc7ebohws8e)

[2.6.7. Trigger giới hạn trùng lặp sách 25](#_h9sqsuukfnzc)

[2.6.8. Trigger giới hạn trùng lặp tên tác giả 26](#_xfeulxmzi0tt)

[2.6.9. Trigger giới hạn trùng lặp nhà xuất bản 26](#_nyjx7gqc5m6l)

[2.6.10. Trigger ngăn mượn sách nếu vi phạm nghiệp vụ 27](#_o064d5ne8gw1)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 29**](#_pspw1wc4cyqe)

[3.1. Kết nối cơ sở dữ liệu 29](#_jcoqu8366la7)

[3.2. Chức năng đăng nhập 33](#_t7ssvs66n5oy)

[3.3. Chức năng đăng ký 33](#_12jjc6xyu3o8)

[3.4. Giao diện của độc giả 34](#_s4q1xo46ysc)

[3.4.1. Tra sách 34](#_f9i1xjd6kris)

[3.4.2. Tra phiếu 36](#_nj6j8il8e43d)

[3.5. Giao diện của quản lý 38](#_remcoq3vr3x4)

[3.5.1. Quản lý tài khoản 38](#_40n4ombopjrm)

[3.5.1.1. Thêm tài khoản 38](#_d5e8tntny93u)

[3.5.1.2. Cập nhật tài khoản 39](#_3cyduj9ypsm)

[3.5.1.3. Tìm tài khoản 40](#_slg3j6wnuzm5)

[3.5.1.4. Xóa tài khoản 41](#_rn0qkrti2ysh)

[3.5.2. Quản lý lịch làm việc 42](#_u6zurp6pf6n8)

[3.5.2.1. Thêm lịch làm việc 42](#_5no0tp82l7as)

[3.5.2.2. Cập nhật lịch làm việc 43](#_ex4wgdip3mp1)

[3.5.2.3. Tìm lịch làm việc 44](#_8ag06debesno)

[3.5.2.4. Xóa lịch làm việc 45](#_7wdsvuhwp4m2)

[3.5.3. Quản lý phiếu mượn 45](#_5p1r2e7917nh)

[3.5.3.1. Thêm phiếu mượn 45](#_rhhmh4xtzu4o)

[3.5.3.2. Tìm phiếu mượn 47](#_cayw6ryyse6c)

[3.5.3.3. Xóa phiếu mượn 47](#_8bczup5m9myk)

[3.5.3.4. Trả phiếu mượn 48](#_omuz46eblfxy)

[3.5.3.5. Cập nhật chi tiết phiếu mượn 51](#_ao7wy5r31hro)

[3.5.4. Quản lý phiếu phạt 52](#_7k742jp1yq7)

[3.5.4.1. Tạo phiếu phạt 52](#_f40x9gqym7tm)

[3.5.4.2. Xóa phiếu phạt 54](#_ynjddwtud3i4)

[3.5.4.3. Thanh toán phiếu phạt 54](#_pn23h1feebc8)

[3.5.4.4. Tìm phiếu phạt 55](#_g3g86hnhzx7x)

[3.5.5. Quản lý sách 56](#_sw3iv92hlpu9)

[3.5.5.1. Thêm sách 56](#_ctlmw624o9g)

[3.5.5.2. Cập nhật sách 57](#_iyefkjizs4xy)

[3.5.5.3. Tìm sách 58](#_nu7u4xvsm70u)

[3.5.5.4. Xoá sách 59](#_p7lgvulpgzx0)

[3.5.6. Quản lý tác giả 60](#_dbbo1shjvlgw)

[3.5.6.1. Thêm tác giả 60](#_nkdam3obyb65)

[3.5.6.2. Cập nhật tác giả 60](#_bwjlzpfdgg0k)

[3.5.6.3. Tìm tác giả 61](#_be2d71gam0rw)

[3.5.6.4. Xoá tác giả 62](#_a2fgsdjdjqr8)

[3.5.7. Quản lý nhà xuất bản 62](#_3bffx6rnkd6w)

[3.5.7.1. Thêm nhà xuất bản 62](#_rs56dcvd467z)

[3.5.7.2. Cập nhật nhà xuất bản 63](#_xrrh3kqhq538)

[3.5.7.3. Tìm nhà xuất bản 64](#_kx6oxwiae7d8)

[3.5.7.4. Xóa nhà xuất bản 64](#_mj9ajsctzjfe)

[3.5.8. Quản lý thể loại 65](#_nr0eivejb6p)

[3.5.8.1. Thêm thể loại 65](#_ga6e6zxy1jcm)

[3.5.8.2. Cập nhật thể loại 66](#_w1npijsiip)

[3.5.8.3. Tìm thể loại 66](#_vcuwiiv7onzw)

[3.5.8.4. Xóa thể loại 67](#_ae117agw6vr)

[3.5.9. Thống kê 68](#_i5i9ess8ewc2)

[3.5.9.1. Thống kê sách 68](#_3iqf8n20m0t6)

[3.5.9.2. Thống kê phiếu 71](#_deze4712a8xu)

[3.5.9.3. Thống kê tài khoản 73](#_nodyhfp1py0m)

[3.6. Chức năng cập nhật tài khoản 76](#_8k6tpcmmvw8i)

[3.7. Chức năng đổi mật khẩu 77](#_raxjlo7rxzda)

[**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 79**](#_8t931tom4esd)

[4.1. Tạo Role trong SQL Server 79](#_wsfnphcyu1lj)

[4.2. Trigger tạo tài khoản SQL Server 80](#_ric3ic6tol2m)

[4.3. Trigger cập nhật tài khoản SQL Server 82](#_yt5jc8tbt9oy)

[4.4. Trigger xóa tài khoản SQL Server 83](#_s2nb2ejw9ejg)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 84**](#_6fw5cttb3ztc)

[5.1. Môi trường cài đặt và công nghệ sử dụng 84](#_xde78vmmpul9)

[5.2. Giao diện và chức năng của chương trình 85](#_6rda1ha8lkkx)

[5.2.1. Form Đăng nhập 85](#_2uso97glbqjq)

[5.2.2. Form Đăng ký 86](#_wi83o4o4g1g8)

[5.2.3. Form Thông tin tài khoản 86](#_ai6nftvvcpad)

[5.2.4. Form Tra sách của độc giả 88](#_wboxnmryjjr6)

[5.2.5. Form Tra phiếu của độc giả 89](#_hut4zpbdn7z0)

[5.2.6. Form Quy định mượn sách thư viện 92](#_py5qa49rmt2g)

[5.2.7. Form Quản lý nhóm tài khoản 93](#_hrfq6wof0msx)

[5.2.8. Form Quản lý lịch làm việc 94](#_d78hmydl9nrd)

[5.2.9. Form Quản lý phiếu mượn 96](#_5fqfjwhos11a)

[5.2.10. Form Quản lý phiếu phạt 97](#_37kbqiudve0r)

[5.2.11. Form Quản lý sách 98](#_ptzvpv92ww73)

[5.2.12. Form Quản lý tác giả 99](#_7bflh157fhi0)

[5.2.13. Form Quản lý nhà xuất bản 100](#_5e8nwy3sid5c)

[5.2.14. Form Quản lý thể loại 101](#_rmsgkfjmn2op)

[5.2.15. Form Thống kê sách 102](#_4ers6oxg8c4r)

[5.2.16. Form Thống kê phiếu mượn và phạt 102](#_uvu5n928k6rb)

[5.2.17. Form Thống kê trên vai trò tài khoản 103](#_1oh61z2kcd1j)

[5.2.18. Form Đổi mật khẩu tài khoản 103](#_2prc2wyqdjdj)

[5.2.19. Form Cập nhật tài khoản 104](#_byea7c42xikn)

[5.2.20. Form Xem chi tiết sách mượn 105](#_q65do1qn65ii)

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

## **1.1. Đặc tả đề tài**

### **1.1.1 Mô tả bài toán**

Một thư viện cần có một hệ thống để quản lý. Chương trình quản lý được xây dựng cho ba đối tượng người dùng là quản trị viên, thủ thư và độc giả với các quyền khác nhau. Đối với người dùng tài khoản với chức vụ là quản trị viên, họ sẽ có quyền truy cập vào mọi chức năng, trong đó các chức năng chỉ dành riêng cho quản trị viên như quản lý nhóm tài khoản, phân làm việc. Đối với người dùng tài khoản với chức vụ là thủ thư, họ sẽ có quyền truy cập vào các chức năng như quản lý sách, quản lý thể loại, quản lý tác giả, quản lý nhà xuất bản và quản lý phiếu mượn, phiếu phạt. Còn với người dùng tài khoản với chức vụ là độc giả, họ sẽ có quyền như đăng ký, đăng nhập và cập nhật tài khoản, tìm kiếm sách, xem chi tiết các phiếu mượn và phiếu phạt. Đối với hệ thống quản lý thư viện này cần lưu trữ những dữ liệu sau:

Một thư viện có nhiều sách, một sách chỉ thuộc về một thư viện, các sách phân biệt với nhau bằng mã sách, sách bao gồm các thuộc tính như: Mã sách, tên sách, mã tác giả, năm xuất bản, số lượng, mã nhà xuất bản, giá sách, loại tài liệu.

Một sách chỉ thuộc về một thể loại, một thể loại có thể có nhiều sách, thể loại gồm các thông tin như: Mã thể loại, tên thể loại.

Một sách chỉ thuộc về một tác giả, một tác giả có thể sáng tác nhiều sách, các tác giả phân biệt với nhau bằng mã tác giả, tác giả bao gồm các thuộc tính như: Mã tác giả, tên tác giả.

Một sách chỉ thuộc về một nhà xuất bản, một nhà xuất bản có thể xuất bản nhiều sách, các nhà xuất bản phân biệt với nhau bằng mã nhà xuất bản, nhà xuất bản bao gồm các thuộc tính như: Mã nhà xuất bản, tên nhà xuất bản.

Mỗi cuốn sách mượn thuộc về một sách và một phiếu mượn, mỗi cuốn sách phân biệt với nhau bằng mã sách và mã phiếu mượn, cuốn sách bao gồm các thuộc tính như: Mã sách, mã phiếu mượn, tình trạng.

Một thư viện có nhiều tài khoản, một tài khoản chỉ thuộc về một thư viện, các tài khoản phân biệt với nhau bằng mã tài khoản, tài khoản bao gồm các thuộc tính như: Mã tài khoản, email, mật khẩu, vai trò, họ tên, số điện thoại, địa chỉ, ngày sinh, giới tính.

Một thủ thư có thể tạo ra nhiều phiếu mượn sách, một phiếu mượn sách chỉ được tạo ra bởi một thủ thư, các phiếu mượn sách phân biệt với nhau bằng mã phiếu mượn, phiếu mượn bao gồm các thuộc tính như: Mã phiếu mượn, mã sách, mã tài khoản, ngày mượn, ngày trả.

Một phiếu mượn sách chỉ thuộc về một độc giả, một độc giả có thể có nhiều phiếu mượn sách dựa trên số lượng giới hạn quy định.

Một thủ thư có thể tạo ra nhiều phiếu phạt, một phiếu phạt chỉ được tạo ra bởi một thủ thư, các phiếu phạt phân biệt với nhau bằng mã phiếu phạt, phiếu phạt bao gồm các thuộc tính như: Mã phiếu phạt, mã phiếu mượn, tiền phạt, ngày trả.

Một độc giả có thể có nhiều phiếu phạt, một phiếu phạt chỉ thuộc về một độc giả.

Lịch làm việc được liên kết với tài khoản, các lịch làm việc phân biệt với nhau bằng mã lịch làm việc. lịch làm việc bao gồm các thuộc tính như: Mã lịch làm việc, ngày làm, ca làm, mã tài khoản,

### **1.1.2. Nghiệp vụ của bài toán**

Bài toán quản lý thư viện sẽ liên quan tới việc quản lý nhân viên thư viện và quản lý người mượn sách trong trường. Đối với người mượn ở trường ĐH SPKT thì sẽ chia thành 4 thành phần (sinh viên đại trà, sinh viên CLC, học viên cao học, giảng viên), thể loại mượn chia thành 2 phần (giáo trình, sách tham khảo)

**Thời gian mượn:**  
Sinh viên đại trà:  
+ Giáo trình: Số lượng: 15;  
 Thời gian: 1 học kỳ;  
+ Sách tham khảo: Số lượng: 10;  
 Thời gian: 3 tuần;

Sinh viên CLC:  
+ Giáo trình: Số lượng: 20;  
 Thời gian: 1 học kỳ;  
+ Sách tham khảo: Số lượng:10;  
 Thời gian: 3 tuần;

Học viên cao học:  
+ Giáo trình: Số lượng: 5;  
 Thời gian: 8 tuần;  
+ Sách tham khảo: Số lượng: 5;  
 Thời gian: 4 tuần;

Giảng viên:  
+ Giáo trình: Số lượng: 5;  
 Thời gian: 1 năm;  
+ Sách tham khảo: Số lượng: 10;  
 Thời gian: 1 năm;

**Quản lý tài khoản:**  
 Quản trị viên có các chức năng đặc biệt như tìm kiếm, tạo, cập nhật và xóa tài khoản.

Quản trị viên, Thủ thư và độc giả đều có thể đăng nhập và cập nhật tài khoản.

Độc giả có thể đăng ký tài khoản.

**Quản lý sách:**

Quản trị viên, thủ thư và độc giả có thể tìm kiếm sách.  
 Chỉ có quản trị viên và thủ thư mới có quyền quản lý sách: thêm sách, xoá sách, cập nhật thông tin sách.

**Quản lý mượn, phạt và trả sách:**  
 Tra thông tin sách hiện có thể mượn thì có thể đăng nhập tài khoản để tra, nếu có vi phạm về vấn đề trả sách thì bị tạo phiếu phạt(Trễ sẽ là 1k/ngày, mất hoặc hư là x2 giá sách). Sách hư, sách mất thì sẽ phải đền bù theo quy định, sinh viên chỉ được xét tốt nghiệp khi không còn nợ sách ở thư viện nhà trường. Mọi tài khoản sẽ xem được số phiếu mượn/phạt.  
 Thủ thư có thể tạo, tra, trả và xóa phiếu mượn/phạt, xem thông tin chi tiết phiếu mượn và phiếu phạt.

**Quản lý tác giả:**  
 Các thông tin của tác giả sẽ được thủ thư tạo, cập nhật và xóa.

**Quản lý nhà xuất bản:**  
 Các thông tin của nhà xuất bản sẽ được thủ thư tạo, cập nhật và xóa.

**Quản lý thể loại:**  
 Các thông tin của thể loại sẽ được thủ thư tạo, cập nhật và xóa.

**Báo cáo thống kê:**  
 Số lượng sách, phiếu mượn/ phạt, tài khoản sẽ được thống kê.

**Phân công:**  
 Thời gian làm việc sẽ được quản trị viên phân công.

### **1.1.3. Mô tả chức năng của bài toán**

* Các quyền hạn của các loại vai trò theo lớn đến bé là: Quản trị viên, thủ thư và độc

giả.

* Quyền hạn cao hơn thì ngoài chức năng riêng chỉ vai trò của mình mới có thì cũng sẽ có mọi chức năng của tất cả các vai trò có quyền hạn thấp hơn.
* Đặc biệt lưu ý ngoài thủ thư và quản trị viên, chỉ có độc giả sử dụng tới chức năng đăng ký tài khoản mới.
* ***Đối với quản trị viên:***

**Quản lý tài khoản:** Tạo, xóa và cập nhật các tài khoản.

**Quản lý lịch làm việc:** Tạo, xóa và cập nhật lịch làm việc cho thủ thư

* ***Đối với thủ thư:***

**Quản lý mượn sách:** Cho phép tạo, xóa và cập nhật thông tin phiếu mượn.

**Quản lý phiếu phạt:** Cho phép tạo, xóa và cập nhật thông tin phiếu phạt.

**Quản lý sách:** Cho phép thêm, xóa, sửa sách của thư viện.

**Quản lý tác giả:** Cho phép thêm, xóa, sửa thông tin các tác giả.

**Quản lý nhà xuất bản:** Cho phép thêm, xóa, sửa thông tin các nhà xuất bản.

**Quản lý thể loại:** Cho phép thêm, xóa, sửa thông tin các thể loại.

* ***Đối với độc giả:***

**Quản lý tài khoản:** Cho phép đăng ký, đăng nhập, đổi mật khẩu và cập nhật thông tin tài khoản.

**Quản lý sách:** Cho tra sách.

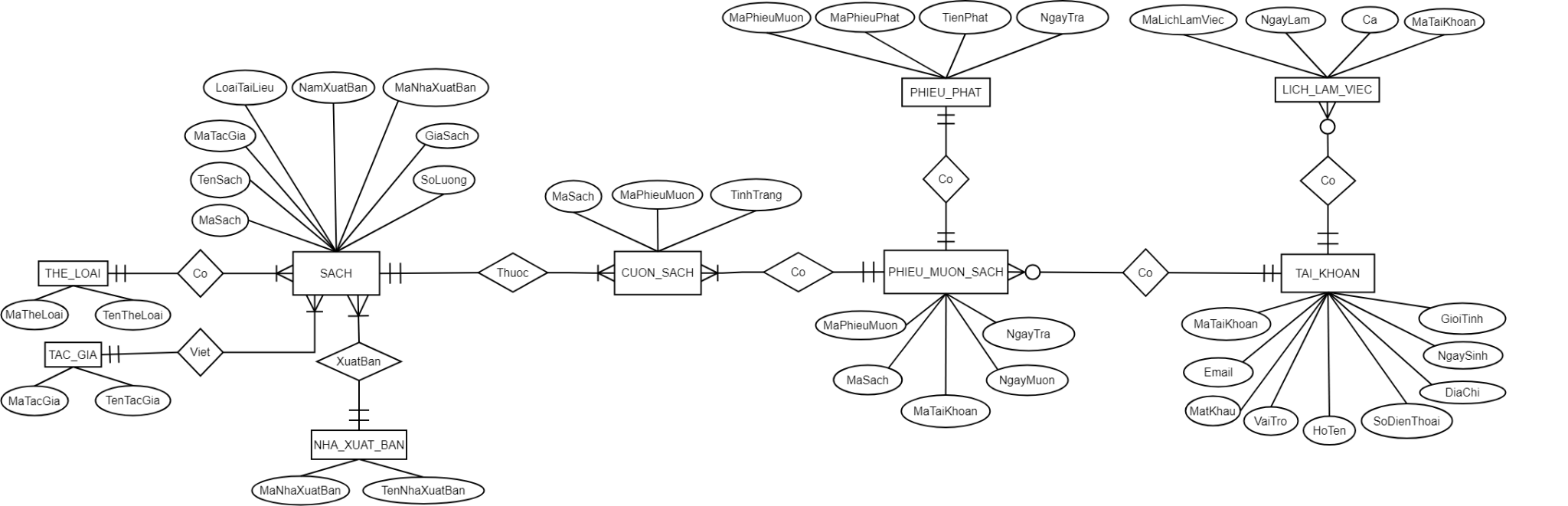
**Quản lý mượn sách:** Cho xem chi tiết phiếu mượn.

**Quản lý phiếu phạt:** Cho xem chi tiết phiếu phạt.

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

## **2.1. Thiết kế cơ sở dữ liệu mức quan niệm**

Từ mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD).

****

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

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* Sach(MaSach, TenSach, MaTacGia, NamXuatBan, TheLoai, SoLuong, MaNhaXuatBan, GiaSach, LoaiTaiLieu)
* TheLoai(MaTheLoai, TenTheLoai)
* TacGia(MaTacGia, TenTacGia)
* NhaXuatBan(MaNhaXuatBan, TenNhaXuatBan, DiaChi)
* TaiKhoan(MaTaiKhoan, Email, MatKhau, VaiTro, HoTen, SoDienThoai, NgaySinh, DiaChi, GioiTinh)
* CuonSach(MaSach, MaPhieuMuon. TinhTrang)
* PhieuMuonSach(MaPhieuMuon, MaSach, MaTaiKhoan, NgayMuon, NgayTra)
* PhieuPhat(MaPhieuPhat, MaPhieuMuon, MaTaiKhoan, MaLoaiTinhTrang, TienPhat)
* LichLamViec(MaLichLamViec, MaTaiKhoan, Ca, NgayLam)

## **2.3. Các ràng buộc cần có**

| STT | Bảng | Ràng buộc |
| --- | --- | --- |
| 1 | Sach | Khoá chính MaSach, Khoá ngoại MaTacGia đến bảng TacGia, Khoá ngoại MaTheLoai đến bảng TheLoai, Khoá ngoại MaNhaXuatBan đến bảng NhaXuatBan, NOT NULL TenSach,  NOT NULL LoaiTaiLieu,  CHECK NamXuatBan(>=0 và <= Năm hiện tại),  CHECK GiaSach(>=0),  CHECK SoLuong(>=0) |
| 2 | TheLoai | Khóa chính MaTheLoai,  NOT NULL TenTheLoai |
| 3 | TacGia | Khoá chính MaTacGia, NOT NULL TenTacGia |
| 4 | NhaXuatBan | Khoá chính MaNhaXuatBan, NOT NULL TenNhaXuatBan |
| 5 | TaiKhoan | Khóa chính MaTaiKhoan,  NOT NULL Email,  CHECK MatKhau(>= 8 ký tự),  NOT NULL VaiTro,  NOT NULL HoTen,  CHECK SoDienThoai(=10),  CHECK NgaySinh(>=18 tuổi),  NOT NULL DiaChi,  NOT NULL GioiTinh |
| 6 | CuonSach | Khóa chính (MaSach, MaPhieuMuon),  Khoá ngoại MaSach đến bảng Sach,  Khóa ngoại MaPhieuMuon đến bảng PhieuMuonSach,  NOT NULL TinhTrang |
| 7 | PhieuMuonSach | Khóa chính MaPhieuMuon,  Khóa ngoại MaSach đến bảng Sach,  Khóa ngoại MaTaiKhoan đến bảng TaiKhoan,  CHECK NgayMuon(= Ngày hiện tại),  NULL NgayTra |
| 8 | PhieuPhat | Khóa chính MaPhieuPhat,  Khóa ngoại MaPhieuMuon đến bảng PhieuMuonSach,  CHECK TienPhat(>=0),  NULL NgayTra |
| 9 | LichLamViec | Khóa chính MaLichLamViec,  CHECK NgayLam(>= Ngày hiện tại),  NOT NULL Ca,  Khóa ngoại MaTaiKhoan đến bảng TaiKhoan |

## **2.4. Cài đặt các CSDL và các ràng buộc**

***Bảng sách***

| CREATE TABLE Sach (  MaSach INT IDENTITY(1,1) PRIMARY KEY,  MaTacGia INT,  MaTheLoai INT,  MaNhaXuatBan INT,  TenSach NVARCHAR(255) NOT NULL,  LoaiTaiLieu NVARCHAR(50) NOT NULL,  NamXuatBan INT CHECK (NamXuatBan >= 0 AND NamXuatBan <= YEAR(GETDATE())),  GiaSach DECIMAL(18,0) CHECK (GiaSach >= 0),  SoLuong INT CHECK (SoLuong >= 0),  FOREIGN KEY (MaTacGia) REFERENCES TacGia(MaTacGia) ON DELETE SET NULL,  FOREIGN KEY (MaTheLoai) REFERENCES TheLoai(MaTheLoai) ON DELETE SET NULL,  FOREIGN KEY (MaNhaXuatBan) REFERENCES NhaXuatBan(MaNhaXuatBan) ON DELETE SET NULL  ); |
| --- |

***Bảng thể loại***

| CREATE TABLE TheLoai(  MaTheLoai INT IDENTITY(1,1) PRIMARY KEY,  TenTheLoai NVARCHAR(255) NOT NULL  ); |
| --- |

***Bảng tác giả***

| CREATE TABLE TacGia (  MaTacGia INT IDENTITY(1,1) PRIMARY KEY,  TenTacGia NVARCHAR(255) NOT NULL  ); |
| --- |

***Bảng nhà xuất bản***

| CREATE TABLE NhaXuatBan (  MaNhaXuatBan IDENTITY(1,1) INT PRIMARY KEY,  TenNhaXuatBan NVARCHAR(255) NOT NULL  ); |
| --- |

***Bảng tài khoản***

| CREATE TABLE TaiKhoan (  MaTaiKhoan INT PRIMARY KEY,  Email NVARCHAR(255) UNIQUE NOT NULL,  MatKhau NVARCHAR(255) CHECK (LEN(MatKhau) >= 8),  VaiTro NVARCHAR(50) NOT NULL,  HoTen NVARCHAR(255) NOT NULL,  SoDienThoai NVARCHAR(10) CHECK (LEN(SoDienThoai) = 10),  NgaySinh DATE CHECK (DATEDIFF(YEAR, NgaySinh, GETDATE()) >= 18),  DiaChi NVARCHAR(255) NOT NULL,  GioiTinh NVARCHAR(10) NOT NULL  ); |
| --- |

***Bảng cuốn sách***

| CREATE TABLE CuonSach(  MaSach INT,  MaPhieuMuon INT,  TinhTrang NVARCHAR(255) NOT NULL,  PRIMARY KEY (MaSach, MaPhieuMuon),  FOREIGN KEY (MaSach) REFERENCES Sach(MaSach) ON DELETE CASCADE,  FOREIGN KEY (MaPhieuMuon) REFERENCES PhieuMuonSach(MaPhieuMuon) ON DELETE CASCADE  ); |
| --- |

***Bảng phiếu mượn sách***

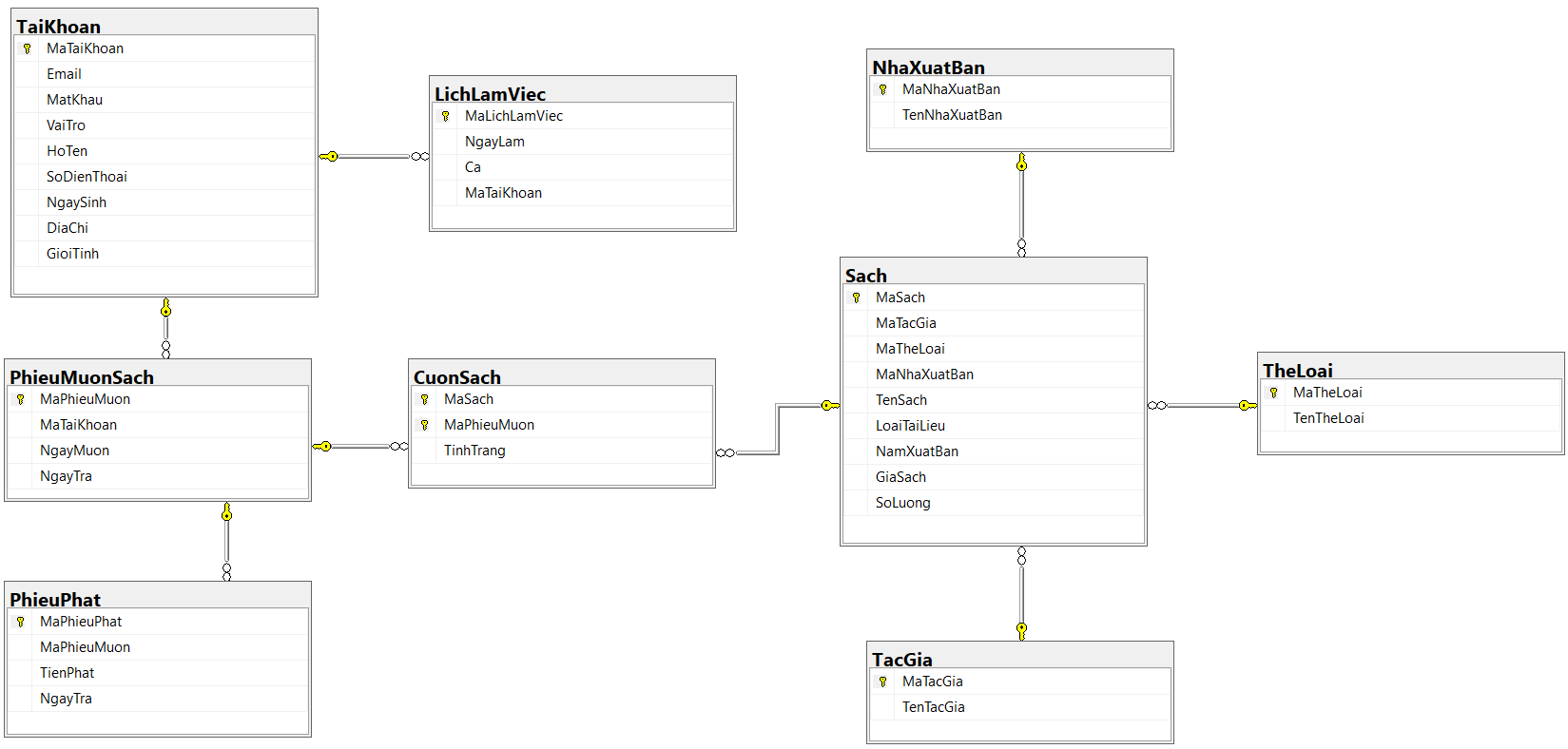
| CREATE TABLE PhieuMuonSach (  MaPhieuMuon INT IDENTITY(1,1) PRIMARY KEY,  MaTaiKhoan INT,  NgayMuon DATE DEFAULT GETDATE(),  CHECK (CONVERT(DATE, NgayMuon) = CONVERT(DATE, GETDATE())),  NgayTra DATE NULL,  FOREIGN KEY (MaTaiKhoan) REFERENCES TaiKhoan(MaTaiKhoan) ON DELETE CASCADE  ); |
| --- |

***Bảng phiếu phạt***

| CREATE TABLE PhieuPhat (  MaPhieuPhat INT IDENTITY(1,1) PRIMARY KEY,  MaPhieuMuon INT,  TienPhat DECIMAL(18,0) CHECK (TienPhat >= 0),  NgayTra DATE NULL,  FOREIGN KEY (MaPhieuMuon) REFERENCES PhieuMuonSach(MaPhieuMuon) ON DELETE CASCADE  ); |
| --- |

***Bảng lịch làm việc***

| CREATE TABLE LichLamViec (  MaLichLamViec INT IDENTITY(1,1) PRIMARY KEY,  NgayLam DATE CHECK (NgayLam >= CAST(GETDATE() AS DATE)),  Ca NVARCHAR(255) NOT NULL,  MaTaiKhoan INT,  FOREIGN KEY (MaTaiKhoan) REFERENCES TaiKhoan(MaTaiKhoan) ON DELETE CASCADE  ); |
| --- |



## **2.5. Các view**

### **2.5.1. View danh sách tài khoản**

| CREATE VIEW VW\_Account\_List AS  SELECT  \*  FROM dbo.TaiKhoan |
| --- |

### **2.5.2. View danh sách lịch làm việc**

| CREATE VIEW VW\_Schedule\_List AS  SELECT  MaLichLamViec, NgayLam, Ca, Tk.MaTaiKhoan, TK.HoTen, TK.SoDienThoai, TK.GioiTinh  FROM dbo.LichLamViec LLV  JOIN dbo.TaiKhoan TK ON TK.MaTaiKhoan = LLV.MaTaiKhoan |
| --- |

### **2.5.3. View danh sách thủ thư**

| CREATE VIEW VW\_Librarian\_List AS  SELECT  MaTaiKhoan,  HoTen  FROM dbo.TaiKhoan  WHERE VaiTro = N'Thủ thư' |
| --- |

### **2.5.4. View danh sách phiếu phạt**

| CREATE VIEW VW\_PhieuPhat\_List AS  SELECT PP.MaPhieuPhat, PP.MaPhieuMuon,PP.NgayTra,PMS.MaTaiKhoan,PMS.NgayTra AS NgayTraSach  FROM dbo.PhieuPhat PP  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon= PP.MaPhieuMuon |
| --- |

### **2.5.5. View danh sách phiếu mượn sách**

| Create VIEW VW\_Book\_Loan\_Coupon\_List AS  SELECT  \*  FROM dbo.PhieuMuonSach |
| --- |

### **2.5.6. View danh sách sách thư viện**

| CREATE VIEW VW\_Book\_List AS  SELECT  S.MaSach,  TG.TenTacGia,  TL.TenTheLoai,  NXB.TenNhaXuatBan,  S.TenSach,  S.LoaiTaiLieu,  S.NamXuatBan,  S.GiaSach,  S.SoLuong  FROM  Sach S  INNER JOIN  TacGia TG ON S.MaTacGia = TG.MaTacGia  INNER JOIN  TheLoai TL ON S.MaTheLoai = TL.MaTheLoai  INNER JOIN  NhaXuatBan NXB ON S.MaNhaXuatBan = NXB.MaNhaXuatBan; |
| --- |

### **2.5.7. View danh sách tác giả**

| CREATE VIEW VW\_TacGia\_List AS  SELECT \*  FROM dbo.TacGia |
| --- |

### **2.5.8. View danh sách nhà xuất bản**

| CREATE VIEW VW\_NhaXuatBan\_List AS  SELECT \*  FROM dbo.NhaXuatBan |
| --- |

### **2.5.9. View danh sách thể loại**

| CREATE VIEW VW\_TheLoai\_List AS  SELECT \*  FROM dbo.TheLoai |
| --- |

## **2.6. Các Trigger**

### **2.6.1. Trigger ngăn bị trùng số điện thoại**

| CREATE TRIGGER TR\_Not\_Duplicate\_Phone\_Number  ON dbo.TaiKhoan  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS  (  SELECT \* FROM inserted i  WHERE EXISTS( SELECT \* FROM dbo.TaiKhoan TK  WHERE (TK.SoDienThoai = i.SoDienThoai)  AND(TK.MaTaiKhoan <> i.MaTaiKhoan)  )  )  BEGIN  RAISERROR ('Duplicate phone number is not allowed', 16, 1);  ROLLBACK TRANSACTION;  END  END; |
| --- |

### **2.6.2. Trigger giới hạn giờ làm việc trong một tuần của thủ thư**

| CREATE TRIGGER TR\_Limit\_Working\_Hours  ON dbo.LichLamViec  AFTER INSERT, UPDATE  AS  BEGIN  DECLARE @MaTaiKhoan INT;  DECLARE @NgayLam Date;  DECLARE @TotalHours INT;  --Một Ca tính là 4 giờ  SELECT @MaTaiKhoan = MaTaiKhoan,  @NgayLam = NgayLam  FROM inserted;    SELECT @TotalHours = dbo.FN\_Total\_Working\_Hours(@MaTaiKhoan, @NgayLam)  IF @TotalHours > 40  BEGIN  RAISERROR ('The total working hours exceed the limit of 40 hours per week for a Librarian.', 16, 1);  ROLLBACK TRANSACTION;  END  END; |
| --- |

Hàm tính tổng thời gian làm việc:

| CREATE FUNCTION FN\_Total\_Working\_Hours  (  @MaTaiKhoan INT,  @NgayLam DATE  )  RETURNS INT  AS  BEGIN  DECLARE @WeekStart DATE;  DECLARE @WeekEnd DATE;  DECLARE @TotalShifts INT;  SET @WeekStart = DATEADD(DAY, 2 - DATEPART(WEEKDAY, @NgayLam), @NgayLam);  SET @WeekEnd = DATEADD(DAY, 8 - DATEPART(WEEKDAY, @NgayLam), @NgayLam);  SELECT @TotalShifts = COUNT(\*)  FROM dbo.LichLamViec  WHERE NgayLam BETWEEN @WeekStart AND @WeekEnd  AND MaTaiKhoan = @MaTaiKhoan;  RETURN @TotalShifts\*4  END; |
| --- |

### **2.6.3. Trigger giới hạn trùng lặp cùng ngày cùng ca làm của một thủ thư**

| CREATE TRIGGER TR\_Not\_Duplicate\_Working\_Days  ON dbo.LichLamViec  AFTER INSERT, UPDATE  AS  BEGIN  DECLARE @MaTaiKhoan INT;  DECLARE @MaLichLamViec INT;  DECLARE @NgayLam Date;  DECLARE @Ca NVARCHAR(255);  SELECT @MaTaiKhoan = MaTaiKhoan, @NgayLam = NgayLam, @Ca = Ca, @MaLichLamViec=MaLichLamViec FROM inserted;    IF EXISTS (SELECT \* FROM dbo.LichLamViec WHERE(MaTaiKhoan = @MaTaiKhoan AND NgayLam = @NgayLam AND Ca = @Ca AND @MaLichLamViec<>MaLichLamViec))  BEGIN  RAISERROR ('A working day can not be duplicate.', 16, 1);  ROLLBACK TRANSACTION;  END  END; |
| --- |

### **2.6.4. Trigger cập nhật số lượng sách khi sau khi tạo phiếu mượn**

| CREATE TRIGGER TR\_Not\_Update\_Books\_Quantity\_After\_Borrowed  ON dbo.CuonSach  AFTER INSERT, UPDATE  AS  BEGIN  DECLARE @MaPhieuMuon INT;  SELECT @MaPhieuMuon = MaPhieuMuon FROM inserted;  UPDATE S  SET S.SoLuong = S.SoLuong - 1  FROM dbo.CuonSach CS  JOIN dbo.Sach S ON CS.MaSach = S.MaSach  WHERE CS.MaPhieuMuon = @MaPhieuMuon;  END; |
| --- |

### **2.6.5. Trigger giới hạn trùng lặp tên thể loại**

| CREATE TRIGGER TR\_Prevent\_duplicate\_category  ON TheLoai  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM TheLoai t  JOIN INSERTED i ON t.TenTheLoai = i.TenTheLoai WHERE t.MaTheLoai<>i.MaTheLoai  )  BEGIN  RAISERROR('Không thể thêm hoặc cập nhật vì đã tồn tại thể loại có cùng tên.', 16, 1);  ROLLBACK TRANSACTION;  END;  END; |
| --- |

### **2.6.6. Trigger giới hạn không cho ngày trả phạt nhỏ hơn ngày trả sách**

| CREATE TRIGGER TR\_Prevent\_return\_date\_update  ON dbo.PhieuPhat  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM INSERTED i  JOIN PhieuMuonSach pms ON i.MaPhieuMuon = pms.MaPhieuMuon  WHERE i.NgayTra < pms.NgayTra  )  BEGIN  RAISERROR('Ngày trả không được lớn hơn ngày trả thực tế.', 16, 1);  ROLLBACK TRANSACTION;  END;  END; |
| --- |

### **2.6.7. Trigger giới hạn trùng lặp sách**

| CREATE TRIGGER TR\_Prevent\_duplicate\_Book  ON dbo.Sach  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM Sach S  JOIN inserted i  ON S.MaTacGia = i.MaTacGia  WHERE S.TenSach = i.TenSach AND  S.MaTacGia = i.MaTacGia AND  S.MaTheLoai = i.MaTheLoai AND  S.MaNhaXuatBan = i.MaNhaXuatBan AND  S.LoaiTaiLieu = i.LoaiTaiLieu AND  S.NamXuatBan = i.NamXuatBan AND  S.GiaSach = i.GiaSach AND  S.MaSach<>i.MaSach)  BEGIN  RAISERROR('Không tạo trùng tựa sách', 16, 1);  ROLLBACK TRANSACTION;  END;  END; |
| --- |

### **2.6.8. Trigger giới hạn trùng lặp tên tác giả**

| CREATE TRIGGER TR\_Prevent\_duplicate\_Author  ON TacGia  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT 1 FROM TacGia TG JOIN inserted i ON TG.MaTacGia = i.MaTacGia  WHERE TG.TenTacGia = i.TenTacGia AND TG.MaTacGia<>i.MaTacGia)  BEGIN  RAISERROR('Không trùng tên tác giả', 16, 1)  ROLLBACK TRANSACTION  END  END |
| --- |

### **2.6.9. Trigger giới hạn trùng lặp nhà xuất bản**

| CREATE or Alter TRIGGER TR\_Prevent\_Publisher\_Duplicate  ON dbo.NhaXuatBan  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT 1 FROM dbo.NhaXuatBan NXB JOIN inserted i ON NXB.MaNhaXuatBan = i.MaNhaXuatBan  WHERE NXB.TenNhaXuatBan = i.TenNhaXuatBan AND NXB.MaNhaXuatBan<>i.MaNhaXuatBan)  BEGIN  RAISERROR('Không trùng tên nhà xuất bản', 16, 1)  ROLLBACK TRANSACTION  END;  END; |
| --- |

### **2.6.10. Trigger ngăn mượn sách nếu vi phạm nghiệp vụ**

| CREATE OR ALTER TRIGGER TR\_Prevent\_Borrowed\_Situations  ON dbo.PhieuMuonSach  AFTER INSERT  AS  BEGIN  DECLARE @ThamKhao INT, @GiaoTrinh INT, @MaTaiKhoan INT, @RollBack BIT = 0;  SELECT @MaTaiKhoan = MaTaiKhoan FROM inserted;  IF EXISTS (  SELECT 1  FROM PhieuPhat pp  JOIN PhieuMuonSach pms ON pp.MaPhieuMuon = pms.MaPhieuMuon  WHERE pms.MaTaiKhoan = @MaTaiKhoan AND pp.NgayTra IS NULL  )  BEGIN  RAISERROR (N'Tài khoản vẫn còn phiếu phạt chưa trả.', 16, 1);  SET @RollBack = 1;  END;  IF EXISTS (  SELECT \*  FROM dbo.PhieuMuonSach PMS  JOIN dbo.CuonSach CS ON CS.MaPhieuMuon = PMS.MaPhieuMuon  WHERE PMS.MaTaiKhoan = @MaTaiKhoan  AND PMS.NgayTra IS NULL  AND CS.TinhTrang = N'Trả trễ'  )  BEGIN  RAISERROR (N'Tài khoản có sách đang trễ hẹn trả.', 16, 1);  SET @RollBack = 1;  END;    SELECT  @ThamKhao = SUM(CASE WHEN S.LoaiTaiLieu = N'Sách tham khảo' THEN 1 ELSE 0 END),  @GiaoTrinh = SUM(CASE WHEN S.LoaiTaiLieu = N'Giáo trình' THEN 1 ELSE 0 END)  FROM dbo.PhieuMuonSach PMS  JOIN inserted i ON i.MaTaiKhoan = PMS.MaTaiKhoan  JOIN dbo.CuonSach CS ON CS.MaPhieuMuon = PMS.MaPhieuMuon  JOIN dbo.Sach S ON S.MaSach = CS.MaSach  IF EXISTS (  SELECT 1  FROM dbo.TaiKhoan TK  JOIN inserted i ON i.MaTaiKhoan = TK.MaTaiKhoan  WHERE  (TK.VaiTro = N'Sinh viên CLC' AND ((@ThamKhao > 10) OR (@GiaoTrinh > 20)))  OR (TK.VaiTro = N'Sinh viên đại trà' AND ((@ThamKhao > 10) OR (@GiaoTrinh > 15)))  OR (TK.VaiTro = N'Học viên cao học' AND ((@ThamKhao > 5) OR (@GiaoTrinh > 5)))  OR (TK.VaiTro = N'Giảng viên' AND ((@ThamKhao > 10) OR (@GiaoTrinh > 5)))  )  BEGIN  RAISERROR (N'Vượt quá số lượng sách được mượn.', 16, 1);  SET @RollBack = 1;  END;  IF(@RollBack = 1)  BEGIN  ROLLBACK TRANSACTION;  END;  END; |
| --- |

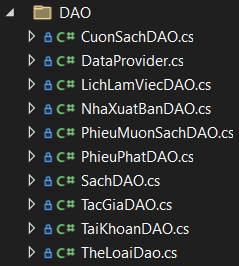
# 

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

## **3.1. Kết nối cơ sở dữ liệu**

| public class DataProvider  {  private static DataProvider instance;  public static DataProvider Instance  {  get { if (instance == null) instance = new DataProvider();  return DataProvider.instance; }  private set { DataProvider.instance = value; }  }  private DataProvider() { }  SqlConnection connection;  private string GetConnectionString()  {  if (DTO.Session.loginAccount == null)  {  return @"Data Source=LAPTOP-M3L0LSMS;Initial  Catalog=Library;Integrated Security=True;TrustServerCertificate=True";  }  return $"Data Source=LAPTOP-M3L0LSMS;Initial Catalog=Library;User ID={DTO.Session.loginAccount.Email};Password={DTO.Session.loginAccount.MatKhau};TrustServerCertificate=True";  }  public DataTable ExecuteQuery(string query, object[] parameter = null)  {  DataTable data = new DataTable();  using (connection = new SqlConnection(GetConnectionString()))  {  connection.Open();  SqlCommand cmd = new SqlCommand(query, connection);  if (parameter != null)  {  string[] listPara = query.Split(' ');  int i = 0;  foreach (string item in listPara)  {  if (item.Contains("@"))  {  cmd.Parameters.AddWithValue(item, parameter[i]);  i++;  }  }  }  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(data);  connection.Close();  }  return data;  }  public int ExecuteNonQuery(string query, object[] parameter = null)  {  int data = 0;  using (connection = new SqlConnection(GetConnectionString()))  {  connection.Open();  SqlCommand cmd = new SqlCommand(query, connection);  if (parameter != null)  {  string[] listPara = query.Split(' ');  int i = 0;  foreach (string item in listPara)  {  if (item.Contains("@"))  {  cmd.Parameters.AddWithValue(item, parameter[i]);  i++;  }  }  }  data = cmd.ExecuteNonQuery();  connection.Close();  }  return data;  }  public object ExecuteScalar(string query, object[] parameter = null)  {  object data = 0;  using (connection = new SqlConnection(GetConnectionString()))  {  connection.Open();  SqlCommand cmd = new SqlCommand(query, connection);  if (parameter != null)  {  string[] listPara = query.Split(' ');  int i = 0;  foreach (string item in listPara)  {  if (item.Contains("@"))  {  cmd.Parameters.AddWithValue(item, parameter[i]);  i++;  }  }  }  data = cmd.ExecuteScalar();  connection.Close();  }  return data;  }  } |
| --- |

Chú thích:



Các lớp trong mục DAO được thiết kế theo mẫu singleton.

Singleton là một mẫu thiết kế trong lập trình giúp tạo ra một thể hiện duy nhất của một lớp và cung cấp một cách truy cập toàn cục đến thể hiện này.

Điểm chính của Singleton là:

-Thể hiện Duy Nhất: Chỉ có một thể hiện của lớp được tạo ra.

-Truy Cập Toàn Cục: Ta có thể truy cập thể hiện của Singleton từ bất kỳ nơi nào trong chương trình.

Ví dụ:

+Lớp bình thường sẽ phải tạo cái mới mỗi khi muốn sử dụng.



+Singleton dùng một cái duy nhất cho cả chương trình.



Kết luận:

-DataProvider Singleton: Cung cấp một thể hiện duy nhất của DataProvider để quản lý kết nối cơ sở dữ liệu.

-Các lớp DAO Singleton: Cung cấp một thể hiện duy nhất của một đối tượng cụ thể để đảm bảo rằng mỗi lần gọi các phương thức từ đối tượng này, bản thân nó sẽ chỉ sử dụng cùng một kết nối từ thể hiện DataProvider Singleton được tạo ra lần đầu khi kết nối với cơ sở dữ liệu và được lưu trữ và mở trong suốt vòng đời của chương trình, giúp tránh xung đột và tối ưu hóa việc sử dụng tài nguyên.

## **3.2. Chức năng đăng nhập**

| CREATE PROCEDURE SP\_Login  @Email NVARCHAR(255),  @MatKhau NVARCHAR(255)  AS  BEGIN  SELECT \* FROM dbo.TaiKhoan WHERE Email = @Email AND MatKhau = @MatKhau  END; |
| --- |

Thực thi trên C#:

| public bool Login(string email, string matKhau)  {  string query = "SP\_Login @Email , @MatKhau ";  DataTable result = DataProvider.Instance.ExecuteQuery(query, new object[] { email, matKhau });  return result.Rows.Count > 0;  } |
| --- |

## **3.3. Chức năng đăng ký**

| CREATE PROCEDURE SP\_Add\_New\_Account  @MaTaiKhoan int,  @HoTen nvarchar(255),  @MatKhau nvarchar(255),  @DiaChi nvarchar(255),  @NgaySinh date,  @Email NVARCHAR(255),  @SoDienThoai NVARCHAR(10),  @VaiTro nvarchar(50),  @GioiTinh nvarchar(10)  AS  BEGIN  BEGIN TRY  INSERT INTO dbo.TaiKhoan (MaTaiKhoan, HoTen, MatKhau, DiaChi, Email, NgaySinh, SoDienThoai, VaiTro, GioiTinh)  VALUES (@MaTaiKhoan, @HoTen, @MatKhau, @DiaChi, @Email, @NgaySinh,@SoDienThoai, @VaiTro, @GioiTinh);  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void AddAccount(TaiKhoan tk)  {  string query = "SP\_Add\_New\_Account @MaTaiKhoan , @HoTen , @MatKhau , @DiaChi , @NgaySinh , @Email , @SoDienThoai , @VaiTro , @GioiTinh ";  DataProvider.Instance.ExecuteQuery(query, new object[] { tk.MaTaiKhoan, tk.HoTen, tk.MatKhau, tk.DiaChi, tk.NgaySinh, tk.Email, tk.SoDienThoai, tk.VaiTro, tk.GioiTinh });  } |
| --- |

## **3.4. Giao diện của độc giả**

### **3.4.1. Tra sách**

| CREATE OR ALTER PROCEDURE SP\_Find\_Book\_By\_Advanced  (  @TenTacGia NVARCHAR(255) = NULL,  @TenTheLoai NVARCHAR(255) = NULL,  @TenNhaXuatBan NVARCHAR(255) = NULL,  @TenSach NVARCHAR(255) = NULL,  @LoaiTaiLieu NVARCHAR(255) = NULL,  @NamXuatBan INT = NULL  )  AS  BEGIN  SELECT \*  FROM VW\_Book\_List  WHERE  (@TenTacGia IS NULL OR @TenTacGia = TenTacGia) AND  (@TenTheLoai IS NULL OR @TenTheLoai = TenTheLoai) AND  (@TenNhaXuatBan IS NULL OR @TenNhaXuatBan = TenNhaXuatBan) AND  (@TenSach IS NULL OR TenSach LIKE '%' + @TenSach + '%') AND  (@LoaiTaiLieu IS NULL OR @LoaiTaiLieu = LoaiTaiLieu) AND  (@NamXuatBan IS NULL OR @NamXuatBan = NamXuatBan)  END; |
| --- |

Thực thi trên C#:

| public DataTable FindBookByAdvanced(Sach s)  {  object tacGia = s.TenTacGia;  object theLoai = s.TenTheLoai;  object nhaXuatBan = s.TenNhaXuatBan;  object tenSach = s.TenSach;  object loaiTaiLieu = s.LoaiTaiLieu;  object namXuatBan = s.NamXuatBan;  if (s.TenTacGia == "")  {  tacGia = DBNull.Value;  }  if (s.TenTheLoai == "")  {  theLoai = DBNull.Value;  }  if (s.TenNhaXuatBan == "")  {  nhaXuatBan = DBNull.Value;  }  if (s.TenSach == "")  {  tenSach = DBNull.Value;  }  if (s.LoaiTaiLieu == "")  {  loaiTaiLieu = DBNull.Value;  }  if (s.NamXuatBan == 0)  {  namXuatBan = DBNull.Value;  }  string query = "SP\_Find\_Book\_By\_Advanced @TenTacGia , @TenTheLoai , @TenNhaXuatBan , @TenSach , @LoaiTaiLieu , @NamXuatBan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { tacGia, theLoai, nhaXuatBan, tenSach, loaiTaiLieu, namXuatBan });  return data;  } |
| --- |

### **3.4.2. Tra phiếu**

| CREATE FUNCTION FN\_Reader\_Borrowed\_List  (  @MaTaiKhoan int,  @Type int -- 0: All, 1: Not Returned, 2: Returned  )  RETURNS TABLE  AS  RETURN  (  SELECT  MaPhieuMuon AS N'Mã phiếu mượn',  NgayMuon AS N'Ngày mượn',  NgayTra AS N'Ngày trả sách'  FROM dbo.PhieuMuonSach  WHERE  MaTaiKhoan = @MaTaiKhoan AND  ((@Type = 0) OR  (@Type = 1 AND NgayTra IS NULL) OR  (@Type = 2 AND NgayTra IS NOT NULL))  ); |
| --- |

| CREATE FUNCTION FN\_Reader\_Penalty\_List  (  @MaTaiKhoan int,  @Type int -- 0: All, 1: Not Paid, 2: Paid  )  RETURNS TABLE  AS  RETURN  (  SELECT  PP.MaPhieuPhat AS N'Mã phiếu phạt',  PP.TienPhat AS N'Tiền phạt',  PP.NgayTra AS N'Ngày đóng phạt'  FROM dbo.PhieuPhat PP  JOIN dbo.PhieuMuonSach PMS ON PP.MaPhieuMuon = PMS.MaPhieuMuon  WHERE  PMS.MaTaiKhoan = @MaTaiKhoan AND  ((@Type = 0) OR  (@Type = 1 AND PP.NgayTra IS NULL) OR  (@Type = 2 AND PP.NgayTra IS NOT NULL))  ); |
| --- |

Thực thi trên C#:

| public DataTable LoadReaderBorrowed(int maTaiKhoan, int type)  {  string query = "SELECT \* FROM FN\_Reader\_Borrowed\_List( @MaTaiKhoan , @type )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan, type });  return data;  } |
| --- |

| public DataTable LoadReaderPenalty(int maTaiKhoan, int type)  {  string query = "SELECT \* FROM FN\_Reader\_Penalty\_List( @MaTaiKhoan , @type )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan, type });  return data;  } |
| --- |

## **3.5. Giao diện của quản lý**

### **3.5.1. Quản lý tài khoản**

#### **3.5.1.**1**. Thêm tài khoản**

| CREATE PROCEDURE SP\_Add\_New\_Account  @MaTaiKhoan int,  @HoTen nvarchar(255),  @MatKhau nvarchar(255),  @DiaChi nvarchar(255),  @NgaySinh date,  @Email NVARCHAR(255),  @SoDienThoai NVARCHAR(10),  @VaiTro nvarchar(50),  @GioiTinh nvarchar(10)  AS  BEGIN  BEGIN TRY  INSERT INTO dbo.TaiKhoan (MaTaiKhoan, HoTen, MatKhau, DiaChi, Email, NgaySinh, SoDienThoai, VaiTro, GioiTinh)  VALUES (@MaTaiKhoan, @HoTen, @MatKhau, @DiaChi, @Email, @NgaySinh,@SoDienThoai, @VaiTro, @GioiTinh);  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void AddAccount(TaiKhoan tk)  {  string query = "SP\_Add\_New\_Account @MaTaiKhoan , @HoTen , @MatKhau , @DiaChi , @NgaySinh , @Email , @SoDienThoai , @VaiTro , @GioiTinh ";  DataProvider.Instance.ExecuteQuery(query, new object[] { tk.MaTaiKhoan, tk.HoTen, tk.MatKhau, tk.DiaChi, tk.NgaySinh, tk.Email, tk.SoDienThoai, tk.VaiTro, tk.GioiTinh });  } |
| --- |

#### **3.5.1.**2**. Cập nhật tài khoản**

| CREATE PROCEDURE SP\_Update\_Account  @MaTaiKhoan int,  @HoTen nvarchar(255),  @MatKhau nvarchar(255),  @DiaChi nvarchar(255),  @NgaySinh date,  @Email nvarchar(255),  @SoDienThoai nvarchar(10),  @VaiTro nvarchar(50),  @GioiTinh nvarchar(10)  AS  BEGIN  BEGIN TRY  UPDATE dbo.TaiKhoan  SET HoTen = @HoTen,  MatKhau = @MatKhau,  DiaChi = @DiaChi,  NgaySinh = @NgaySinh,  GioiTinh = @GioiTinh,  Email = @Email,  SoDienThoai = @SoDienThoai,  VaiTro = @VaiTro  WHERE MaTaiKhoan = @MaTaiKhoan;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public bool UpdateAccount(TaiKhoan tk)  {  string query = "SP\_Update\_Account @MaTaiKhoan , @HoTen , @MatKhau , @DiaChi , @NgaySinh , @Email , @SoDienThoai , @VaiTro , @GioiTinh ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { tk.MaTaiKhoan, tk.HoTen, tk.MatKhau, tk.DiaChi, tk.NgaySinh, tk.Email, tk.SoDienThoai, tk.VaiTro, tk.GioiTinh });  return result > 0;  } |
| --- |

#### **3.5.1.**3**. Tìm tài khoản**

| CREATE OR ALTER PROCEDURE SP\_Find\_Account\_By\_Advanced  (  @MaTaiKhoan int = NULL,  @HoTen nvarchar(255) = NULL,  @DiaChi nvarchar(255) = NULL,  @Email NVARCHAR(255) = NULL,  @SoDienThoai NVARCHAR(10) = NULL,  @VaiTro nvarchar(50) = NULL,  @GioiTinh nvarchar(10) = NULL  )  AS  BEGIN  SELECT \*  FROM VW\_Account\_List  WHERE  (@MaTaiKhoan IS NULL OR MaTaiKhoan = @MaTaiKhoan) AND  (@HoTen IS NULL OR HoTen LIKE '%' + @HoTen + '%') AND  (@DiaChi IS NULL OR DiaChi LIKE '%' + @DiaChi + '%') AND  (@Email IS NULL OR Email LIKE '%' + @Email + '%') AND  (@SoDienThoai IS NULL OR CAST(SoDienThoai AS VARCHAR) LIKE '%' + CAST(@SoDienThoai AS VARCHAR) + '%') AND  (@VaiTro IS NULL OR VaiTro = @VaiTro) AND  (@GioiTinh IS NULL OR GioiTinh = @GioiTinh)  END; |
| --- |

Thực thi trên C#:

| public DataTable FindAccountByAdvanced(TaiKhoan tk)  {  object maTaiKhoan = tk.MaTaiKhoan;  object hoTen = tk.HoTen;  object email = tk.Email;  object vaiTro = tk.VaiTro;  object gioiTinh = tk.GioiTinh;  object diaChi = tk.DiaChi;  object soDienThoai = tk.SoDienThoai;  if (tk.MaTaiKhoan == 0)  {  maTaiKhoan = DBNull.Value;  }  if (tk.HoTen == "")  {  hoTen = DBNull.Value;  }  if (tk.Email == "")  {  email = DBNull.Value;  }  if (tk.VaiTro == null)  {  vaiTro = DBNull.Value;  }  if (tk.GioiTinh == null)  {  gioiTinh = DBNull.Value;  }  if (tk.DiaChi == "")  {  diaChi = DBNull.Value;  }  if (tk.SoDienThoai == "")  {  soDienThoai = DBNull.Value;  }  string query = "SP\_Find\_Account\_By\_Advanced @MaTaiKhoan , @HoTen , @DiaChi , @Email , @SoDienThoai , @VaiTro , @GioiTinh ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan, hoTen, diaChi, email, soDienThoai, vaiTro, gioiTinh });  return data;  } |
| --- |

#### **3.5.1.**4**. Xóa tài khoản**

| CREATE PROCEDURE SP\_Delete\_Account  @MaTaiKhoan int  AS  BEGIN  BEGIN TRY  DELETE dbo.TaiKhoan WHERE MaTaiKhoan = @MaTaiKhoan;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteAccount(int maTaiKhoan)  {  string query = "SP\_Delete\_Account @MaTaiKhoan ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan });  } |
| --- |

### **3.5.2. Quản lý lịch làm việc**

#### **3.5.2.1. Thêm lịch làm việc**

| CREATE OR ALTER PROCEDURE SP\_Add\_New\_Schedule  (  @NgayLam date,  @Ca nvarchar(255),  @MaTaiKhoan int  )  AS  BEGIN  BEGIN TRY  INSERT INTO dbo.LichLamViec(NgayLam, Ca, MaTaiKhoan)  VALUES (@NgayLam , @Ca , @MaTaiKhoan );  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void AddNewSchedule(LichLamViec lich)  {  string query = "SP\_Add\_New\_Schedule @NgayLam , @Ca , @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { lich.NgayLam, lich.Ca, lich.MaTaiKhoan});  } |
| --- |

#### **3.5.2.2. Cập nhật lịch làm việc**

| CREATE OR ALTER PROCEDURE SP\_Update\_Schedule  (  @MaLichLamViec int,  @NgayLam date,  @Ca nvarchar(255),  @MaTaiKhoan int  )  AS  BEGIN  BEGIN TRY  Update dbo.LichLamViec  SET NgayLam = @NgayLam,  Ca = @Ca ,  MaTaiKhoan = @MaTaiKhoan  WHERE MaLichLamViec = @MaLichLamViec  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void UpdateSchedule(LichLamViec lich)  {  string query = "SP\_Update\_Schedule @MaLichLamViec , @NgayLam , @Ca , @MaTaiKhoan ";  DataProvider.Instance.ExecuteQuery(query, new object[] { lich.MaLichLamViec, lich.NgayLam, lich.Ca, lich.MaTaiKhoan }); } |
| --- |

#### **3.5.2.3. Tìm lịch làm việc**

| CREATE OR ALTER PROCEDURE SP\_Get\_Schedule  (  @NgayDauTuan date = NULL,  @NgayCuoiTuan date = NULL,  @MaTaiKhoan int = NULL  )  AS  BEGIN  SELECT \* FROM VW\_Schedule\_List  WHERE  (@NgayDauTuan IS NULL OR NgayLam >= @NgayDauTuan) AND  (@NgayCuoiTuan IS NULL OR NgayLam <= @NgayCuoiTuan) AND  (@MaTaiKhoan IS NULL OR MaTaiKhoan = @MaTaiKhoan)  END; |
| --- |

Thực thi trên C#:

| public DataTable LoadLichLamViecList(DateTime weekstart, DateTime weekend, string maTaiKhoan = null)  {  if (maTaiKhoan == null)  {  string query = "SP\_Get\_Schedule @NgayDauTuan , @NgayCuoiTuan , @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { weekstart, weekend, DBNull.Value });  return data;  }  else  {  int id = Convert.ToInt32(maTaiKhoan);  string query = "SP\_Get\_Schedule @NgayDauTuan , @NgayCuoiTuan , @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { weekstart, weekend, id });  return data;  }  } |
| --- |

#### **3.5.2.4. Xóa lịch làm việc**

| CREATE OR ALTER PROCEDURE SP\_Delete\_Schedule  (  @MaLichLamViec int  )  AS  BEGIN  BEGIN TRY  DELETE dbo.LichLamViec WHERE MaLichLamViec = @MaLichLamViec  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteSchedule(int id)  {  string query = "SP\_Delete\_Schedule @MaLichLamViec ";  DataProvider.Instance.ExecuteQuery(query, new object[] { id });  } |
| --- |

### **3.5.3. Quản lý phiếu mượn**

#### **3.5.3.1. Thêm phiếu mượn**

| CREATE OR ALTER PROCEDURE SP\_Add\_New\_Book\_Loan\_Coupon  @MaTaiKhoan int = NULL,  @NgayMuon date = NULL,  @NgayTra date = NULL  AS  BEGIN  BEGIN TRY  DECLARE @InsertedID INT;  INSERT INTO dbo.PhieuMuonSach (MaTaiKhoan, NgayMuon, NgayTra)  VALUES (@MaTaiKhoan, @NgayMuon, @NgayTra);    SET @InsertedID = SCOPE\_IDENTITY();  SELECT @InsertedID AS InsertedID;    END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public int AddBookLoanCoupon(PhieuMuonSach pm)  {  string query = "SP\_Add\_New\_Book\_Loan\_Coupon @MaTaiKhoan , @NgayMuon , @NgayTra ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { pm.MaTaiKhoan, pm.NgayMuon, DBNull.Value } );  if (data.Rows.Count > 0)  {  int insertedId = (int)data.Rows[0]["InsertedID"];  return insertedId;  }  else  {  return -1;  }  } |
| --- |

#### **3.5.3.2. Tìm phiếu mượn**

| CREATE OR ALTER PROCEDURE SP\_Find\_Account\_Book\_Loan\_Coupon  (  @MaTaiKhoan int  )  AS  BEGIN  SELECT \*  FROM dbo.PhieuMuonSach  WHERE MaTaiKhoan = @MaTaiKhoan  END; |
| --- |

Thực thi trên C#:

| public DataTable FindBookLoanCoupon(int id)  {  string query = "SP\_Find\_Account\_Book\_Loan\_Coupon @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { id });  return data;  } |
| --- |

#### **3.5.3.3. Xóa phiếu mượn**

| CREATE OR ALTER PROCEDURE SP\_Delete\_Coupon  @MaPhieuMuon int  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  UPDATE S  SET S.SoLuong = S.SoLuong + 1  FROM dbo.CuonSach CS  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon = CS.MaPhieuMuon  JOIN dbo.Sach S ON CS.MaSach = S.MaSach  WHERE CS.MaPhieuMuon = @MaPhieuMuon AND NgayTra IS NULL;  DELETE FROM dbo.PhieuMuonSach  WHERE MaPhieuMuon = @MaPhieuMuon    COMMIT;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  ROLLBACK;  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteCoupon(int id)  {  string query = "SP\_Delete\_Coupon @MaPhieuMuon ";  DataProvider.Instance.ExecuteQuery(query, new object[] { id });  } |
| --- |

#### 3.5.3.4. Trả phiếu mượn

| CREATE OR ALTER PROCEDURE SP\_Update\_Coupon\_Returned  @MaPhieuMuon INT  AS  BEGIN  IF EXISTS (  SELECT 1  FROM dbo.CuonSach CS  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon = CS.MaPhieuMuon  JOIN dbo.Sach S ON CS.MaSach = S.MaSach  JOIN dbo.TaiKhoan TK ON TK.MaTaiKhoan = PMS.MaTaiKhoan  WHERE CS.MaPhieuMuon = @MaPhieuMuon  AND (CS.TinhTrang = N'Đang mượn')  AND (  (TK.VaiTro = N'Sinh viên CLC' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 21)  OR (S.Loai TaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 105)  ))  OR  (TK.VaiTro = N'Sinh viên đại trà' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 21)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 105)  ))  OR  (TK.VaiTro = N'Học viên cao học' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 32)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 56)  ))  OR  (TK.VaiTro = N'Giảng viên' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 365)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 365)  ))  )  )  BEGIN  UPDATE CS  SET TinhTrang = N'Trả trễ'  FROM dbo.CuonSach CS  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon = CS.MaPhieuMuon  JOIN dbo.Sach S ON CS.MaSach = S.MaSach  JOIN dbo.TaiKhoan TK ON TK.MaTaiKhoan = PMS.MaTaiKhoan  WHERE CS.MaPhieuMuon = @MaPhieuMuon  AND CS.TinhTrang = N'Đang mượn'  AND (  (TK.VaiTro = N'Sinh viên CLC' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 21)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 105)  ))  OR  (TK.VaiTro = N'Sinh viên đại trà' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 21)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 105)  ))  OR  (TK.VaiTro = N'Học viên cao học' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 32)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 56)  ))  OR  (TK.VaiTro = N'Giảng viên' AND (  (S.LoaiTaiLieu = N'Sách tham khảo' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 365)  OR (S.LoaiTaiLieu = N'Giáo trình' AND DATEDIFF(DAY, PMS.NgayMuon, GETDATE()) > 365)  ))  )  RAISERROR(N'Người này đang giữ sách quá hạn trả.', 16, 1)  RETURN;  END  BEGIN TRANSACTION;  BEGIN TRY  UPDATE S  SET S.SoLuong = S.SoLuong + 1  FROM dbo.CuonSach CS  JOIN dbo.Sach S ON CS.MaSach = S.MaSach  WHERE CS.MaPhieuMuon = @MaPhieuMuon AND (CS.TinhTrang = N'Đã trả' OR CS.TinhTrang = N'Trả trễ');  UPDATE dbo.PhieuMuonSach  SET NgayTra = GETDATE()  WHERE MaPhieuMuon = @MaPhieuMuon;  COMMIT;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  ROLLBACK;  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void UpdateCoupon\_Returned(int id)  {  string query = "SP\_Update\_Coupon\_Returned @MaPhieuMuon ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { id });  } |
| --- |

#### 3.5.3.5. Cập nhật chi tiết phiếu mượn

| CREATE OR ALTER PROCEDURE SP\_Update\_Book\_In\_Coupon  (  @MaPhieuMuon int,  @MaSach int,  @TinhTrang nvarchar(255)  )  AS  BEGIN  BEGIN TRY  UPDATE dbo.CuonSach  SET TinhTrang = @TinhTrang  WHERE MaSach = @MaSach AND MaPhieuMuon = @MaPhieuMuon  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void UpdateStatus(int maPhieuMuon, int maSach, string tinhTrang)  {  string query = "SP\_Update\_Book\_In\_Coupon @MaPhieuMuon , @MaSach , @TinhTrang ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maPhieuMuon, maSach, tinhTrang });  } |
| --- |

#### 

### **3.5.4. Quản lý phiếu phạt**

#### **3.5.4.1.** Tạo **phiếu phạt**

| CREATE OR ALTER PROCEDURE SP\_Add\_New\_PhieuPhat  @MaPhieuMuon int  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  UPDATE dbo.PhieuMuonSach  SET NgayTra = GETDATE()  WHERE MaPhieuMuon = @MaPhieuMuon  UPDATE dbo.CuonSach  SET TinhTrang = N'Đã trả'  WHERE MaPhieuMuon = @MaPhieuMuon AND TinhTrang = N'Đang mượn'  DECLARE @TienPhat DECIMAL(18, 0) = dbo.FN\_Calculate\_Penalty\_Value(@MaPhieuMuon);  INSERT INTO dbo.PhieuPhat (MaPhieuMuon, TienPhat)  VALUES (@MaPhieuMuon, @TienPhat);    COMMIT;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  ROLLBACK;  END CATCH;  END; |
| --- |

| CREATE OR ALTER FUNCTION FN\_Calculate\_Penalty\_Value  (  @MaPhieuMuon INT  )  RETURNS DECIMAL(18, 0)  AS  BEGIN  DECLARE @Penalty DECIMAL(18, 0) = 0;  DECLARE @LateBooks INT = 0;  DECLARE @TotalDays INT = 0;  DECLARE @TotalDamagedOrLost DECIMAL(18, 0) = 0;  SELECT @TotalDamagedOrLost = ISNULL(SUM(S.GiaSach), 0)  FROM dbo.PhieuMuonSach PMS  JOIN dbo.CuonSach CS ON CS.MaPhieuMuon = PMS.MaPhieuMuon  JOIN dbo.Sach S ON S.MaSach = CS.MaSach  WHERE CS.MaPhieuMuon = @MaPhieuMuon AND (CS.TinhTrang = N'Đã hư' OR CS.TinhTrang = N'Đã mất');  SET @TotalDamagedOrLost = @TotalDamagedOrLost \* 2;  IF EXISTS (SELECT 1 FROM dbo.PhieuMuonSach PMS  JOIN dbo.CuonSach CS ON CS.MaPhieuMuon = PMS.MaPhieuMuon  WHERE CS.MaPhieuMuon = @MaPhieuMuon AND CS.TinhTrang = N'Trả trễ')  BEGIN  SELECT @LateBooks = COUNT(\*)  FROM dbo.PhieuMuonSach PMS  JOIN dbo.CuonSach CS ON CS.MaPhieuMuon = PMS.MaPhieuMuon  WHERE CS.MaPhieuMuon = @MaPhieuMuon;  END;  SELECT @TotalDays = DATEDIFF(DAY, PMS.NgayTra, GETDATE())  FROM dbo.PhieuMuonSach PMS  WHERE PMS.MaPhieuMuon = @MaPhieuMuon;  SET @Penalty = CAST(ISNULL(@TotalDays, 0) AS DECIMAL(18, 0)) \* ISNULL(@LateBooks, 0) \* 1000 + ISNULL(@TotalDamagedOrLost, 0);  RETURN @Penalty;  END; |
| --- |

Thực thi trên C#:

| public void AddPhieuPhat(PhieuPhat pp)  {  string query = "SP\_Add\_New\_PhieuPhat @MaPhieuMuon ";  DataProvider.Instance.ExecuteQuery(query, new object[] { pp.MaPhieuMuon });  } |
| --- |

#### 3.5.4.2. Xóa phiếu phạt

| CREATE OR ALTER PROCEDURE SP\_Delete\_PhieuPhat  @MaPhieuPhat int  AS  BEGIN  BEGIN TRY  DELETE dbo.PhieuPhat WHERE MaPhieuPhat = @MaPhieuPhat;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeletePhieuPhat(int maPhieuPhat)  {  string query = "SP\_Delete\_PhieuPhat @MaPhieuPhat ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maPhieuPhat });  } |
| --- |

#### 3.5.4.3. Thanh toán phiếu phạt

| CREATE OR ALTER PROCEDURE SP\_Pay\_Penalty\_Coupon\_Debt  @MaPhieuPhat int  AS  BEGIN  BEGIN TRY  UPDATE dbo.PhieuPhat  SET NgayTra = GETDATE()  WHERE MaPhieuPhat = @MaPhieuPhat  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void PayDebt(int maPhieuPhat)  {  string query = "SP\_Pay\_Penalty\_Coupon\_Debt @MaPhieuPhat ";  DataProvider.Instance.ExecuteNonQuery(query, new object[] { maPhieuPhat });  } |
| --- |

#### 3.5.4.4. Tìm phiếu phạt

| CREATE OR ALTER PROCEDURE SP\_Find\_PhieuPhat  (  @MaTaiKhoan int  )  AS  BEGIN  SELECT  PP.MaPhieuPhat, PP.MaPhieuMuon, PP.TienPhat, PP.NgayTra  FROM dbo.PhieuPhat PP  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon = PP.MaPhieuMuon  WHERE PMS.MaTaiKhoan = @MaTaiKhoan  END; |
| --- |

Thực thi trên C#:

| public DataTable FindPhieuPhat(int id)  {  string query = "SP\_Find\_PhieuPhat @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { id });  return data;  } |
| --- |

### **3.5.5. Quản lý sách**

#### 3.5.5.1. Thêm sách

| CREATE PROC SP\_Add\_New\_Book  @MaTacGia INT,  @MaTheLoai INT,  @MaNhaXuatBan INT,  @TenSach NVARCHAR(255),  @LoaiTaiLieu NVARCHAR(255),  @NamXuatBan INT,  @GiaSach DECIMAL,  @SoLuong INT  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO dbo.Sach (MaTacGia, MaTheLoai, MaNhaXuatBan, TenSach, LoaiTaiLieu, NamXuatBan, GiaSach, SoLuong)  VALUES (@MaTacGia, @MaTheLoai, @MaNhaXuatBan, @TenSach, @LoaiTaiLieu, @NamXuatBan, @GiaSach, @SoLuong);    COMMIT;  END; |
| --- |

Thực thi trên C#:

| public void AddBook(Sach s)  {  string query = "SP\_Add\_New\_Book @MaTacGia , @MaTheLoai , @MaNhaXuatBan , @TenSach , @LoaiTaiLieu , @NamXuatBan , @GiaSach , @SoLuong ";  DataProvider.Instance.ExecuteQuery(query, new object[] { s.MaTacGia, s.MaTheLoai, s.MaNhaXuatBan, s.TenSach, s.LoaiTaiLieu, s.NamXuatBan, s.GiaSach, s.SoLuong });  } |
| --- |

#### 3.5.5.2. Cập nhật sách

| CREATE PROC SP\_Update\_Book  @MaSach INT,  @MaTacGia INT,  @MaTheLoai INT,  @MaNhaXuatBan INT,  @TenSach NVARCHAR(255),  @LoaiTaiLieu NVARCHAR(255),  @NamXuatBan INT,  @GiaSach DECIMAL,  @SoLuong INT  AS  BEGIN  BEGIN TRANSACTION;  UPDATE dbo.Sach  SET MaTacGia = @MaTacGia,  MaTheLoai = @MaTheLoai,  MaNhaXuatBan = @MaNhaXuatBan,  TenSach = @TenSach,  LoaiTaiLieu = @LoaiTaiLieu,  NamXuatBan = @NamXuatBan,  GiaSach = @GiaSach,  SoLuong = @SoLuong  WHERE MaSach = @MaSach;  COMMIT;  END; |
| --- |

Thực thi trên C#:

| public int UpdateBook(Sach s)  {  string query = "SP\_Update\_Book @MaSach , @MaTacGia , @MaTheLoai , @MaNhaXuatBan , @TenSach , @LoaiTaiLieu , @NamXuatBan , @GiaSach , @SoLuong ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { s.MaSach, s.MaTacGia, s.MaTheLoai, s.MaNhaXuatBan, s.TenSach, s.LoaiTaiLieu, s.NamXuatBan, s.GiaSach, s.SoLuong });  return result;  } |
| --- |

#### 3.5.5.3. Tìm sách

| CREATE OR ALTER PROCEDURE SP\_Find\_Book\_By\_Advanced  (  @TenTacGia NVARCHAR(255) = NULL,  @TenTheLoai NVARCHAR(255) = NULL,  @TenNhaXuatBan NVARCHAR(255) = NULL,  @TenSach NVARCHAR(255) = NULL,  @LoaiTaiLieu NVARCHAR(255) = NULL,  @NamXuatBan INT = NULL  )  AS  BEGIN  SELECT \*  FROM VW\_Book\_List  WHERE  (@TenTacGia IS NULL OR @TenTacGia = TenTacGia) AND  (@TenTheLoai IS NULL OR @TenTheLoai = TenTheLoai) AND  (@TenNhaXuatBan IS NULL OR @TenNhaXuatBan = TenNhaXuatBan) AND  (@TenSach IS NULL OR TenSach LIKE '%' + @TenSach + '%') AND  (@LoaiTaiLieu IS NULL OR @LoaiTaiLieu = LoaiTaiLieu) AND  (@NamXuatBan IS NULL OR @NamXuatBan = NamXuatBan)  END; |
| --- |

Thực thi trên C#:

| public DataTable FindBookByAdvanced(Sach s)  {  object tacGia = s.TenTacGia;  object theLoai = s.TenTheLoai;  object nhaXuatBan = s.TenNhaXuatBan;  object tenSach = s.TenSach;  object loaiTaiLieu = s.LoaiTaiLieu;  object namXuatBan = s.NamXuatBan;  if (s.TenTacGia == "")  {  tacGia = DBNull.Value;  }  if (s.TenTheLoai == "")  {  theLoai = DBNull.Value;  }  if (s.TenNhaXuatBan == "")  {  nhaXuatBan = DBNull.Value;  }  if (s.TenSach == "")  {  tenSach = DBNull.Value;  }  if (s.LoaiTaiLieu == "")  {  loaiTaiLieu = DBNull.Value;  }  if (s.NamXuatBan == 0)  {  namXuatBan = DBNull.Value;  }  string query = "SP\_Find\_Book\_By\_Advanced @TenTacGia , @TenTheLoai , @TenNhaXuatBan , @TenSach , @LoaiTaiLieu , @NamXuatBan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { tacGia, theLoai, nhaXuatBan, tenSach, loaiTaiLieu, namXuatBan });  return data;  } |
| --- |

#### 3.5.5.4. Xoá sách

| CREATE PROC SP\_Delete\_Book  @MaSach int  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  DELETE dbo.Sach WHERE MaSach = @MaSach;    COMMIT;  END TRY  BEGIN CATCH  ROLLBACK;  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteBook(int maSach)  {  string query = "SP\_Delete\_Book @MaSach ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maSach });  } |
| --- |

### **3.5.6. Quản lý tác giả**

#### 3.5.6.1. Thêm tác giả

| CREATE PROC SP\_Add\_Tac\_Gia  @TenTacGia NVARCHAR(255)  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO dbo.TacGia (TenTacGia)  VALUES (@TenTacGia);    COMMIT;  END; |
| --- |

Thực thi trên C#:

| public void AddTacGia(TacGia tg)  {  string query = "SP\_Add\_Tac\_Gia @TenTacGia ";  DataProvider.Instance.ExecuteQuery(query, new object[] { tg.TenTacGia });  } |
| --- |

#### 3.5.6.2. Cập nhật tác giả

| CREATE PROC SP\_Update\_Tac\_Gia  @MaTacGia INT,  @TenTacGia NVARCHAR(255)  AS  BEGIN  BEGIN TRANSACTION;  UPDATE dbo.TacGia  SET TenTacGia = @TenTacGia  WHERE MaTacGia = @MaTacGia;  COMMIT;  END; |
| --- |

Thực thi trên C#:

| public bool UpdateTacGia(TacGia tg)  {  string query = "SP\_Update\_Tac\_Gia @MaTacGia , @TenTacGia ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { tg.MaTacGia, tg.TenTacGia });  return result > 0;  } |
| --- |

#### 3.5.6.3. Tìm tác giả

| CREATE OR ALTER PROCEDURE SP\_Find\_TacGia  (  @TenTacGia NVARCHAR(255) = NULL  )  AS  BEGIN  SELECT  \*  FROM VW\_TacGia\_List WHERE @TenTacGia LIKE '%' + TenTacGia + '%'    END; |
| --- |

Thực thi trên C#:

| public DataTable FindTacGiaByAdvanced(TacGia tg)  {  object tenTacGia = tg.TenTacGia;  if (tg.TenTacGia == "")  {  tenTacGia = DBNull.Value;  }  string query = "SP\_Find\_TacGia @TenTacGia ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { tenTacGia });  return data;  } |
| --- |

#### 3.5.6.4. Xoá tác giả

| CREATE PROC SP\_Delete\_TacGia  @MaTacGia int  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  DELETE dbo.TacGia WHERE MaTacGia = @MaTacGia;    COMMIT;  END TRY  BEGIN CATCH  ROLLBACK;  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteTacGia(int maTacGia)  {  string query = "SP\_Delete\_TacGia @MaTacGia ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maTacGia });  } |
| --- |

### **3.5.7. Quản lý nhà xuất bản**

#### 3.5.7.1. Thêm nhà xuất bản

| CREATE OR ALTER PROCEDURE SP\_Add\_New\_NXB  @TenNhaXuatBan NVARCHAR(255)  AS  BEGIN  BEGIN TRY  INSERT INTO dbo.NhaXuatBan(TenNhaXuatBan)  VALUES (@TenNhaXuatBan);  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void AddNXB(NhaXuatBan nxb)  {  string query = "SP\_Add\_New\_NXB @TenNhaXuatBan ";  DataProvider.Instance.ExecuteQuery(query, new object[] { nxb.TenNhaXuatBan });  } |
| --- |

#### 3.5.7.2. Cập nhật nhà xuất bản

| CREATE OR ALTER PROCEDURE SP\_Update\_NXB  @MaNhaXuatBan INT,  @TenNhaXuatBan NVARCHAR(255)  AS  BEGIN  BEGIN TRY  UPDATE dbo.NhaXuatBan  Set TenNhaXuatBan = @TenNhaXuatBan  Where MaNhaXuatBan = @MaNhaXuatBan;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public bool UpdateNXB(NhaXuatBan nxb)  {  string query = "SP\_Update\_NXB @MaNhaXuatBan , @TenNhaXuatBan ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { nxb.MaNhaXuatBan, nxb.TenNhaXuatBan });  return result > 0;  } |
| --- |

#### 3.5.7.3. Tìm nhà xuất bản

| CREATE OR ALTER PROCEDURE SP\_Find\_NXB  (  @TenNhaXuatBan NVARCHAR(255) = NULL  )  AS  BEGIN  SELECT \*  FROM VW\_NXB\_List  WHERE TenNhaXuatBan LIKE '%' + @TenNhaXuatBan + '%'  END; |
| --- |

Thực thi trên C#:

| public DataTable FindNhaXuatBan(string tenNXB)  {  object nxb = tenNXB;  if (tenNXB == "")  {  nxb = DBNull.Value;  }  string query = "SP\_Find\_NXB @TenNhaXuatBan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { nxb });  return data;  } |
| --- |

#### 3.5.7.4. Xóa nhà xuất bản

| CREATE OR ALTER PROCEDURE SP\_Delete\_NXB  @MaNhaXuatBan INT,  @TenNhaXuatBan NVARCHAR(255)  AS  BEGIN  BEGIN TRY  Delete From dbo.NhaXuatBan  Where MaNhaXuatBan = @MaNhaXuatBan  And TenNhaXuatBan = @TenNhaXuatBan;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteNXB(NhaXuatBan nxb)  {  string query = "SP\_Delete\_NXB @MaNhaXuatBan , @TenNhaXuatBan ";  DataProvider.Instance.ExecuteQuery(query, new object[] { nxb.MaNhaXuatBan, nxb.TenNhaXuatBan });  } |
| --- |

### **3.5.8. Quản lý thể loại**

#### 3.5.8.1. Thêm thể loại

| CREATE OR ALTER PROCEDURE SP\_Add\_New\_TheLoai  @TenTheLoai nvarchar(50)  AS  BEGIN  BEGIN TRY  INSERT INTO dbo.TheLoai (TenTheLoai)  VALUES (@TenTheLoai);  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void AddTheLoai(TheLoai tl)  {  string query = "SP\_Add\_New\_TheLoai @TenTheLoai ";  DataProvider.Instance.ExecuteQuery(query, new object[] { tl.TenTheLoai });  } |
| --- |

#### 3.5.8.2. Cập nhật thể loại

| CREATE OR ALTER PROCEDURE SP\_Update\_TheLoai  @MaTheLoai int,  @TenTheLoai nvarchar(50)  AS  BEGIN  BEGIN TRY  UPDATE dbo.TheLoai  SET TenTheLoai = @TenTheLoai  WHERE MaTheLoai = @MaTheLoai;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public bool UpdateTheLoai(TheLoai tl)  {  string query = "SP\_Update\_TheLoai @MaTheLoai , @TenTheLoai ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { tl.MaTheLoai, tl.TenTheLoai });  return result > 0;  } |
| --- |

#### 3.5.8.3. Tìm thể loại

| CREATE OR ALTER PROCEDURE SP\_Find\_TheLoai  (  @TenTheLoai nvarchar(255) = NULL  )  AS  BEGIN  SELECT \* FROM VW\_TheLoai\_List WHERE TenTheLoai LIKE '%' + @TenTheLoai + '%'  END; |
| --- |

Thực thi trên C#:

| public DataTable FindTheLoai(string tenTheLoai)  {  object TenTheLoai = tenTheLoai;  if (tenTheLoai == "")  {  TenTheLoai = DBNull.Value;  }  string query = "SP\_Find\_TheLoai @TenTheLoai ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { TenTheLoai });  return data;  } |
| --- |

#### 3.5.8.4. Xóa thể loại

| CREATE OR ALTER PROCEDURE SP\_Delete\_TheLoai  @MaTheLoai int  AS  BEGIN  BEGIN TRY  DELETE dbo.TheLoai WHERE MaTheLoai = @MaTheLoai;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public void DeleteTheLoai(int maTheLoai)  {  string query = "SP\_Delete\_TheLoai @MaTheLoai ";  DataProvider.Instance.ExecuteQuery(query, new object[] { maTheLoai });  } |
| --- |

### **3.5.9. Thống kê**

#### 3.5.9.1. Thống kê sách

Tổng số lượng tác giả:

| CREATE FUNCTION FN\_Total\_Authors()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Authors  FROM dbo.TacGia  ); |
| --- |

| public DataTable TotalAuthors()  {  string query = "SELECT \* FROM FN\_Total\_Authors()";  DataTable total = DataProvider.Instance.ExecuteQuery(query);  return total;  } |
| --- |

Tổng số lượng thể loại:

| CREATE FUNCTION FN\_Total\_Categories()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Categories  FROM dbo.TheLoai  ); |
| --- |

Thực thi trên C#:

| public DataTable TotalCategories()  {  string query = "SELECT \* FROM FN\_Total\_Categories()";  DataTable total = DataProvider.Instance.ExecuteQuery(query);  return total;  } |
| --- |

Tổng số lượng nhà xuất bản:

| CREATE FUNCTION FN\_Total\_Publishers()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Publishers  FROM dbo.NhaXuatBan  ); |
| --- |

Thực thi trên C#:

| public DataTable TotalPublishers()  {  string query = "SELECT \* FROM FN\_Total\_Publishers()";  DataTable total = DataProvider.Instance.ExecuteQuery(query);  return total;  } |
| --- |

Tổng số lượng sách:

| CREATE FUNCTION FN\_Total\_Books()  RETURNS TABLE  AS  RETURN (  SELECT  (SELECT SUM(Total\_Available) FROM FN\_Total\_Available\_Books()) +  (SELECT SUM(Total\_Borrowed) FROM FN\_Total\_Borrowed\_Books()) +  (SELECT SUM(Total\_Damaged\_Or\_Lost) FROM FN\_Total\_Damaged\_Or\_Lost\_Books()) AS Total\_All  ); |
| --- |

Thực thi trên C#:

| public int TotalBooks()  {  string query = "SELECT \* FROM FN\_Total\_Books()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng sách có sẵn:

| CREATE FUNCTION FN\_Total\_Available\_Books()  RETURNS TABLE  AS  RETURN (  SELECT SUM(DISTINCT SoLuong) AS Total\_Available  FROM dbo.Sach  ); |
| --- |

Thực thi trên C#:

| public int TotalAvailableBooks()  {  string query = "SELECT \* FROM FN\_Total\_Available\_Books()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng sách đã mượn:

| CREATE FUNCTION FN\_Total\_Borrowed\_Books()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Borrowed  FROM dbo.CuonSach  WHERE TinhTrang = N'Đang mượn'  ); |
| --- |

Thực thi trên C#:

| public int TotalBorrowedBooks()  {  string query = "SELECT \* FROM FN\_Total\_Borrowed\_Books()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng sách bị mất hoặc bị hư:

| CREATE FUNCTION FN\_Total\_Damaged\_Or\_Lost\_Books()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Damaged\_Or\_Lost  FROM dbo.CuonSach  WHERE TinhTrang = N'Đã hư' OR TinhTrang = N'Đã mất'  ); |
| --- |

Thực thi trên C#:

| public int TotalDamagedOrLostBooks()  {  string query = "SELECT \* FROM FN\_Total\_Damaged\_Or\_Lost\_Books()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

#### 3.5.9.2. Thống kê phiếu

Tổng số lượng phiếu mượn theo tháng:

| CREATE FUNCTION FN\_Total\_Loan\_Coupons\_By\_Month  (  @month int,  @year int  )  RETURNS TABLE  AS  RETURN (  SELECT  COUNT(\*) AS Total\_Loan\_Coupons  FROM  dbo.PhieuMuonSach  WHERE  MONTH(NgayMuon) = @month AND YEAR(NgayMuon) = @year  ); |
| --- |

Thực thi trên C#:

| public int TotalLoanCouponsByMonth(int month, int year)  {  string query = "SELECT \* FROM dbo.FN\_Total\_Loan\_Coupons\_By\_Month( @month , @year )";  object total = DataProvider.Instance.ExecuteScalar(query, new object[] { month, year });  return (int)total;  } |
| --- |

Tổng số lượng phiếu phạt theo tháng:

| CREATE FUNCTION FN\_Total\_Penalty\_Coupons\_By\_Month  (  @month int,  @year int  )  RETURNS TABLE  AS  RETURN (  SELECT  COUNT(\*) AS Total\_Penalty\_Coupons  FROM  dbo.PhieuPhat  WHERE  MONTH(NgayTra) = @month AND YEAR(NgayTra) = @year  ); |
| --- |

Thực thi trên C#:

| public int TotalPenaltyCouponsByMonth(int month, int year)  {  string query = "SELECT \* FROM dbo.FN\_Total\_Penalty\_Coupons\_By\_Month( @month , @year )";  object total = DataProvider.Instance.ExecuteScalar(query, new object[] { month, year });  return (int)total;  } |
| --- |

#### 3.5.9.3. Thống kê tài khoản

Tổng số lượng tài khoản:

| CREATE FUNCTION FN\_Total\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Accounts  FROM dbo.TaiKhoan  ); |
| --- |

Thực thi trên C#:

| public int TotalAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của sinh viên chất lượng cao:

| CREATE FUNCTION FN\_Total\_High\_Quality\_Student\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_High\_Quality\_Student\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Sinh viên CLC'  ); |
| --- |

Thực thi trên C#:

| public int TotalHighQualityStudentAccounts()  {  string query = "SELECT \* FROM FN\_Total\_High\_Quality\_Student\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của sinh viên đại trà:

| CREATE FUNCTION FN\_Total\_Mass\_Student\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Mass\_Student\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Sinh viên đại trà'  ); |
| --- |

Thực thi trên C#:

| public int TotalMassStudentAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Mass\_Student\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của sinh viên cao học:

| CREATE FUNCTION FN\_Total\_Graduate\_Student\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Graduate\_Student\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Học viên cao học'  ); |
| --- |

Thực thi trên C#:

| public int TotalGraduateStudentAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Graduate\_Student\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của giảng viên:

| CREATE FUNCTION FN\_Total\_Lecturer\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Lecturer\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Giảng viên'  ); |
| --- |

Thực thi trên C#:

| public int TotalLecturerAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Lecturer\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của thủ thư:

| CREATE FUNCTION FN\_Total\_Librarian\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Librarian\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Thủ thư'  ); |
| --- |

Thực thi trên C#

| public int TotalLibrarianAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Librarian\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

Tổng số lượng tài khoản của quản trị viên:

| CREATE FUNCTION FN\_Total\_Administrator\_Accounts()  RETURNS TABLE  AS  RETURN (  SELECT COUNT(\*) AS Total\_Administrator\_Accounts  FROM dbo.TaiKhoan  WHERE VaiTro = N'Quản trị viên'  ); |
| --- |

Thực thi trên C#

| public int TotalAdministratorAccounts()  {  string query = "SELECT \* FROM FN\_Total\_Administrator\_Accounts()";  object total = DataProvider.Instance.ExecuteScalar(query);  return (int)total;  } |
| --- |

## **3.6. Chức năng cập nhật tài khoản**

| CREATE PROCEDURE SP\_Update\_Account  @MaTaiKhoan int,  @HoTen nvarchar(255),  @MatKhau nvarchar(255),  @DiaChi nvarchar(255),  @NgaySinh date,  @Email nvarchar(255),  @SoDienThoai nvarchar(10),  @VaiTro nvarchar(50),  @GioiTinh nvarchar(10)  AS  BEGIN  BEGIN TRY  UPDATE dbo.TaiKhoan  SET HoTen = @HoTen,  MatKhau = @MatKhau,  DiaChi = @DiaChi,  NgaySinh = @NgaySinh,  GioiTinh = @GioiTinh,  Email = @Email,  SoDienThoai = @SoDienThoai,  VaiTro = @VaiTro  WHERE MaTaiKhoan = @MaTaiKhoan;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END; |
| --- |

Thực thi trên C#:

| public bool UpdateProfile(TaiKhoan tk)  {  string query = "SP\_Update\_Account @MaTaiKhoan , @HoTen , @DiaChi , @NgaySinh , @SoDienThoai , @GioiTinh ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { tk.MaTaiKhoan, tk.HoTen, tk.DiaChi, tk.NgaySinh, tk.SoDienThoai, tk.GioiTinh });  return result > 0;  } |
| --- |

### 

## **3.7. Chức năng đổi mật khẩu**

| CREATE PROCEDURE SP\_Change\_Account\_Password  (  @Email NVARCHAR(255),  @MatKhauMoi NVARCHAR(255),  @XacNhan NVARCHAR(255),  @MatKhauCu NVARCHAR(255)  )  AS  BEGIN  IF NOT EXISTS (SELECT \* FROM dbo.TaiKhoan WHERE Email = @Email AND MatKhau = @MatKhauCu)  BEGIN  RAISERROR('Email hoặc mật khẩu cũ không đúng', 16, 1);  RETURN;  END  IF @MatKhauMoi < > @XacNhan  BEGIN  RAISERROR('Mật khẩu mới và xác nhận không khớp', 16, 1);  RETURN;  END  BEGIN TRY  UPDATE dbo.TaiKhoan  SET MatKhau = @MatKhauMoi  WHERE Email = @Email;  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH;  END |
| --- |

Thực hiện trên C#:

| public bool ChangeAccountPassword(string email, string newPass, string confirm, string oldPass)  {  string query = "SP\_Change\_Account\_Password @Email , @MatKhauMoi , @XacNhan , @MatKhauCu ";  int result = DataProvider.Instance.ExecuteNonQuery(query, new object[] { email, newPass, confirm, oldPass });  return result > 0;  } |
| --- |

# 

# **CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

## **4.1. Tạo Role trong SQL Server**

| Quản trị viên role là sysadmin  CREATE ROLE ThuThu;  -- Grant permissions on tables  GRANT SELECT, REFERENCES ON [TaiKhoan] TO ThuThu;  GRANT SELECT ON [LichLamViec] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON [NhaXuatBan] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON [TacGia] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON [TheLoai] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON [Sach] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON [PhieuMuonSach] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE ON [PhieuPhat] TO ThuThu;  GRANT SELECT, INSERT, UPDATE, DELETE ON [CuonSach] TO ThuThu;  --Views + Functions  GRANT SELECT TO ThuThu  --Procedures  GRANT EXECUTE TO ThuThu  DENY EXECUTE ON [SP\_Add\_New\_Account] TO ThuThu  DENY EXECUTE ON [SP\_Delete\_Account] TO ThuThu  DENY EXECUTE ON [SP\_Update\_Account] TO ThuThu  DENY EXECUTE ON [SP\_Add\_New\_Schedule] TO ThuThu  DENY EXECUTE ON [SP\_Update\_Schedule] TO ThuThu  DENY EXECUTE ON [SP\_Delete\_Schedule] TO ThuThu  CREATE ROLE DocGia;  -- Grant permissions on tables  GRANT SELECT, REFERENCES ON [TaiKhoan] TO DocGia;  GRANT SELECT, REFERENCES ON [NhaXuatBan] TO DocGia;  GRANT SELECT, REFERENCES ON [TacGia] TO DocGia;  GRANT SELECT, REFERENCES ON [TheLoai] TO DocGia;  GRANT SELECT, REFERENCES ON [Sach] TO DocGia;  GRANT SELECT, REFERENCES ON [PhieuMuonSach] TO DocGia;  GRANT SELECT ON [PhieuPhat] TO DocGia;  GRANT SELECT ON [CuonSach] TO DocGia;  --Views  GRANT SELECT ON [VW\_Book\_List] TO DocGia  GRANT SELECT ON [VW\_NhaXuatBan\_List] TO DocGia  GRANT SELECT ON [VW\_TheLoai\_List] TO DocGia  GRANT SELECT ON [VW\_TacGia\_List] TO DocGia  GRANT SELECT ON [VW\_NXB\_List] TO DocGia  --Procedures + Functions  GRANT EXECUTE ON [SP\_Login] TO DocGia  GRANT EXECUTE ON [SP\_Change\_Account\_Password] TO DocGia  GRANT EXECUTE ON [SP\_Account\_Profile] TO DocGia  GRANT EXECUTE ON [SP\_Find\_Book\_By\_Advanced] TO DocGia  GRANT EXECUTE ON [SP\_Find\_TacGia] TO DocGia  GRANT EXECUTE ON [SP\_Find\_NXB] TO DocGia  GRANT EXECUTE ON [SP\_Find\_TheLoai] TO DocGia  GRANT SELECT ON [FN\_Reader\_Borrowed\_Detail] TO DocGia  GRANT SELECT ON [FN\_Reader\_Penalty\_List] TO DocGia  GRANT SELECT ON [FN\_Reader\_Borrowed\_List] TO DocGia |
| --- |

## **4.2. Trigger tạo tài khoản SQL Server**

| CREATE TRIGGER TR\_Create\_SQL\_Server\_Account  ON dbo.TaiKhoan  AFTER INSERT  AS  BEGIN  DECLARE @Email NVARCHAR(255),  @MatKhau NVARCHAR(255),  @VaiTro NVARCHAR(255),  @MaTaiKhoan NVARCHAR(255),  @Query NVARCHAR(MAX);  SELECT @Email = Email,  @MatKhau = MatKhau,  @VaiTro = VaiTro,  @MaTaiKhoan = MaTaiKhoan  FROM inserted;  SET @Query = 'CREATE LOGIN [' + @Email + '] WITH PASSWORD = N''' + @MatKhau + ''', DEFAULT\_DATABASE=[Library], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF;';  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error creating LOGIN: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  SET @Query = 'CREATE USER [' + @Email + '] FOR LOGIN [' + @Email + '];';  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error creating USER: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  IF (@VaiTro = N'Quản trị viên')  BEGIN  SET @Query = 'ALTER SERVER ROLE sysadmin ADD MEMBER [' + @Email + '];';  END  ELSE IF (@VaiTro = N'Thủ thư')  BEGIN  SET @Query = 'ALTER ROLE ThuThu ADD MEMBER [' + @Email + '];';  END  ELSE  BEGIN  SET @Query = 'ALTER ROLE DocGia ADD MEMBER [' + @Email + '];';  END  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error assigning role: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  END; |
| --- |

## **4.3. Trigger cập nhật tài khoản SQL Server**

| CREATE TRIGGER TR\_Update\_SQL\_Server\_Account  ON dbo.TaiKhoan  AFTER UPDATE  AS  BEGIN  DECLARE @Email NVARCHAR(255),  @MatKhau NVARCHAR(255),  @VaiTro NVARCHAR(255),  @MaTaiKhoan NVARCHAR(255),  @OldEmail NVARCHAR(255),  @OldMatKhau NVARCHAR(255),  @Query NVARCHAR(MAX);  SELECT @Email = Email,  @MatKhau = MatKhau,  @VaiTro = VaiTro,  @MaTaiKhoan = MaTaiKhoan  FROM inserted;  SELECT @OldEmail = Email,  @OldMatKhau = MatKhau  FROM deleted;  IF (@Email <> @OldEmail OR @MatKhau <> @OldMatKhau)  BEGIN  SET @Query = 'ALTER LOGIN [' + @OldEmail + '] WITH PASSWORD = N''' + @MatKhau + ''', NAME = [' + @Email + '];';  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error updating LOGIN: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  SET @Query = 'ALTER USER [' + @OldEmail + '] WITH NAME = [' + @Email + '];';  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error updating USER: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  END  IF (@VaiTro = N'Quản trị viên')  BEGIN  SET @Query = 'ALTER SERVER ROLE sysadmin ADD MEMBER [' + @Email + '];';  END  ELSE IF (@VaiTro = N'Thủ thư')  BEGIN  SET @Query = 'ALTER ROLE ThuThu ADD MEMBER [' + @Email + '];';  END  ELSE  BEGIN  SET @Query = 'ALTER ROLE DocGia ADD MEMBER [' + @Email + '];';  END  BEGIN TRY  EXEC sp\_executesql @Query;  END TRY  BEGIN CATCH  PRINT 'Error assigning role: ' + ERROR\_MESSAGE();  ROLLBACK TRANSACTION;  END CATCH;  END; |
| --- |

## **4.4. Trigger xóa tài khoản SQL Server**

| CREATE TRIGGER TR\_Delete\_SQL\_Server\_Account  ON dbo.TaiKhoan  AFTER DELETE  AS  BEGIN  DECLARE @Email NVARCHAR(255),  @Query NVARCHAR(MAX),  @SessionID INT;  SELECT @Email = Email  FROM deleted;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @Email;  IF @SessionID IS NOT NULL  BEGIN  SET @Query = 'KILL ' + CONVERT(nvarchar(20), @SessionID);  EXEC (@Query);  END  SET @Query = 'DROP USER [' + @Email + '];';  EXEC (@Query);  SET @Query = 'DROP LOGIN [' + @Email + '];';  EXEC (@Query);  END; |
| --- |

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

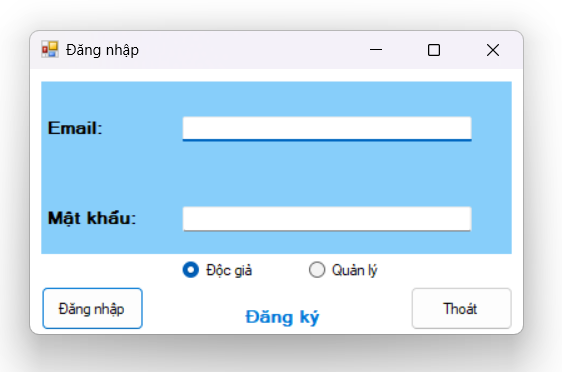
## **5.1. Môi trường cài đặt và công nghệ sử dụng**

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

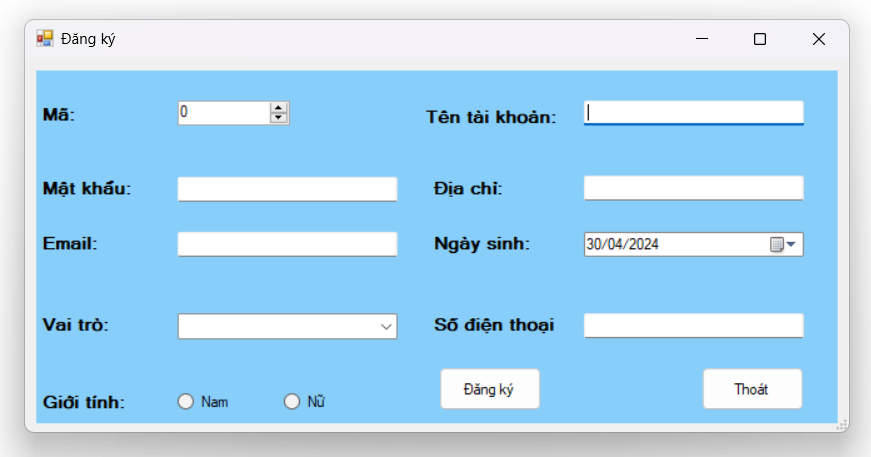
-Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

## **5.2. Giao diện và chức năng của chương trình**

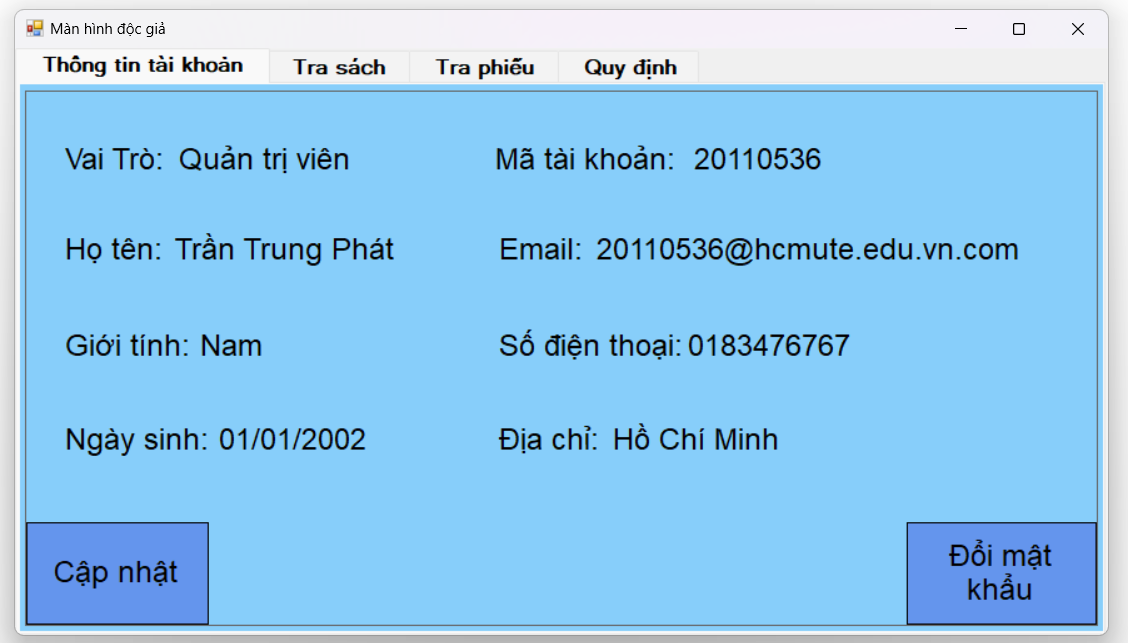
### **5.2.1. Form Đăng nhập**



### **5.2.2. Form Đăng ký**



### **5.2.3. Form Thông tin tài khoản**



### 

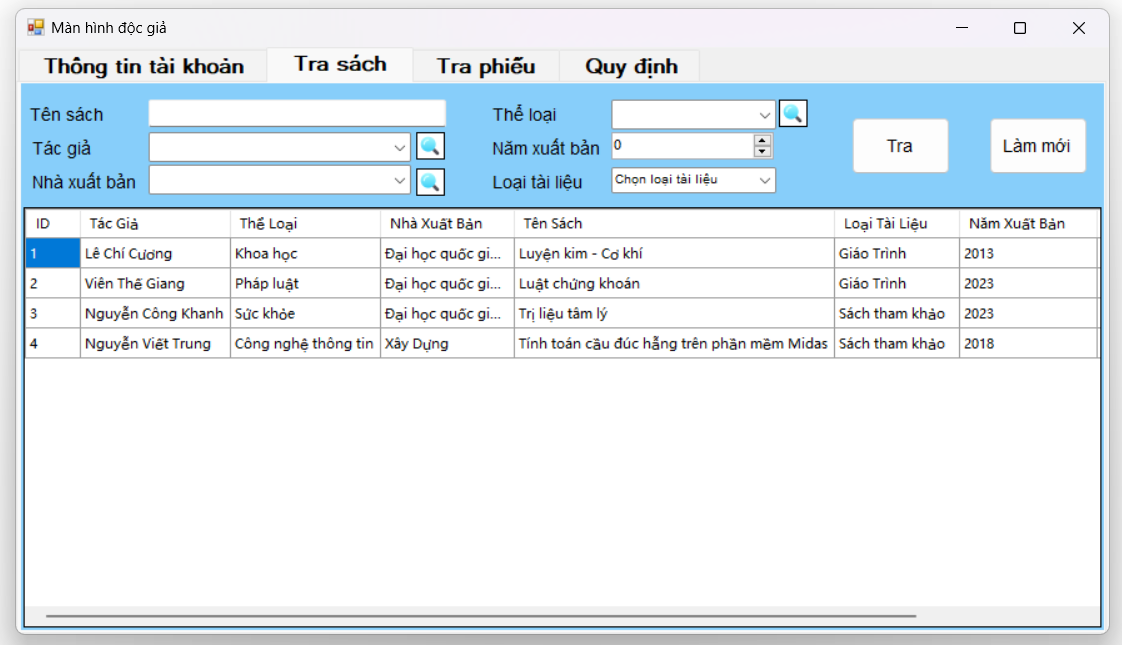
| CREATE OR ALTER PROCEDURE SP\_Get\_Account\_Profile  (@Email nvarchar(255))  AS  BEGIN  SELECT \*  FROM TaiKhoan  WHERE Email = @Email  END; |
| --- |

Thực thi trên C#:

| public TaiKhoan GetAccountProfile(string email)  {  string query = "SP\_Get\_Account\_Profile @Email ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { email });  foreach (DataRow item in data.Rows)  {  return new TaiKhoan(item);  }  return null;  } |
| --- |

### 

### **5.2.4. Form Tra sách của độc giả**



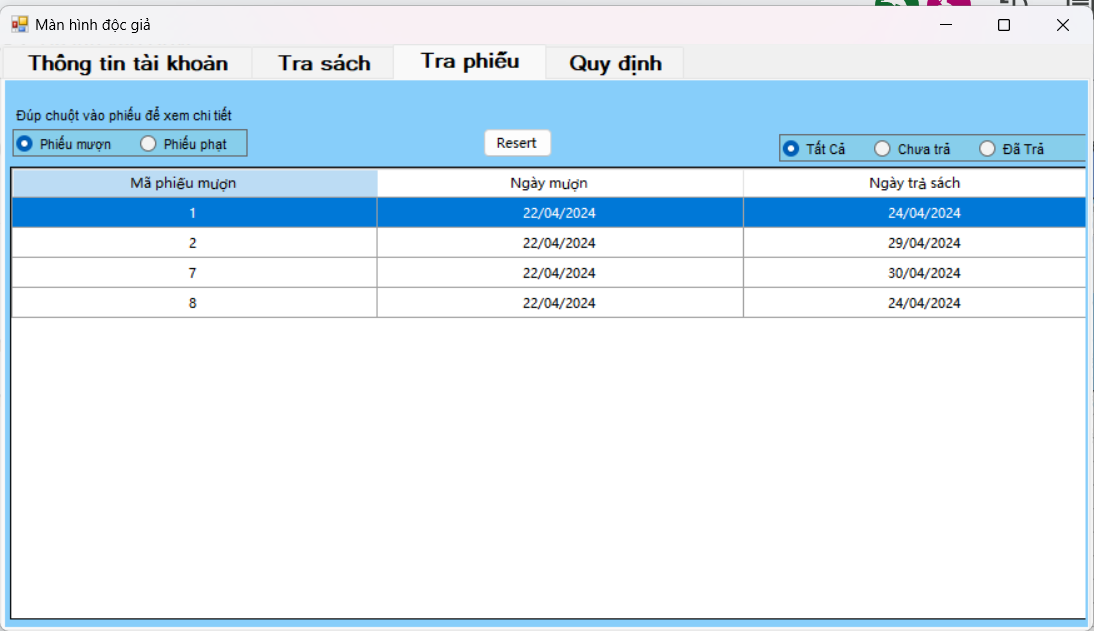
| CREATE VIEW VW\_Book\_List AS  SELECT  S.MaSach,  TG.TenTacGia,  TL.TenTheLoai,  NXB.TenNhaXuatBan,  S.TenSach,  S.LoaiTaiLieu,  S.NamXuatBan,  S.GiaSach,  S.SoLuong  FROM  Sach S  INNER JOIN  TacGia TG ON S.MaTacGia = TG.MaTacGia  INNER JOIN  TheLoai TL ON S.MaTheLoai = TL.MaTheLoai  INNER JOIN  NhaXuatBan NXB ON S.MaNhaXuatBan = NXB.MaNhaXuatBan; |
| --- |

Thực thi trên C#:

| public DataTable LoadBookList()  {  string query = "SELECT \* FROM VW\_Book\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### 

### **5.2.5. Form Tra phiếu của độc giả**



Tra phiếu mượn

| CREATE FUNCTION FN\_Reader\_Borrowed\_List  (  @MaTaiKhoan int,  @Type int -- 0: All, 1: Not Returned, 2: Returned  )  RETURNS TABLE  AS  RETURN  (  SELECT  MaPhieuMuon AS N'Mã phiếu mượn',  NgayMuon AS N'Ngày mượn',  NgayTra AS N'Ngày trả sách'  FROM dbo.PhieuMuonSach  WHERE  MaTaiKhoan = @MaTaiKhoan AND  ((@Type = 0) OR  (@Type = 1 AND NgayTra IS NULL) OR  (@Type = 2 AND NgayTra IS NOT NULL))  ); |
| --- |

Thực thi trên C#:

| public DataTable LoadReaderBorrowed(int maTaiKhoan, int type)  {  string query = "SELECT \* FROM FN\_Reader\_Borrowed\_List( @MaTaiKhoan , @type )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan, type });  return data;  } |
| --- |

Tra phiếu phạt

| CREATE FUNCTION FN\_Reader\_Penalty\_List  (  @MaTaiKhoan int,  @Type int -- 0: All, 1: Not Paid, 2: Paid  )  RETURNS TABLE  AS  RETURN  (  SELECT  PP.MaPhieuPhat AS N'Mã phiếu phạt',  PP.TienPhat AS N'Tiền phạt',  PP.NgayTra AS N'Ngày đóng phạt'  FROM dbo.PhieuPhat PP  JOIN dbo.PhieuMuonSach PMS ON PP.MaPhieuMuon = PMS.MaPhieuMuon  WHERE  PMS.MaTaiKhoan = @MaTaiKhoan AND  ((@Type = 0) OR  (@Type = 1 AND PP.NgayTra IS NULL) OR  (@Type = 2 AND PP.NgayTra IS NOT NULL))  ); |
| --- |

Thực thi trên C#:

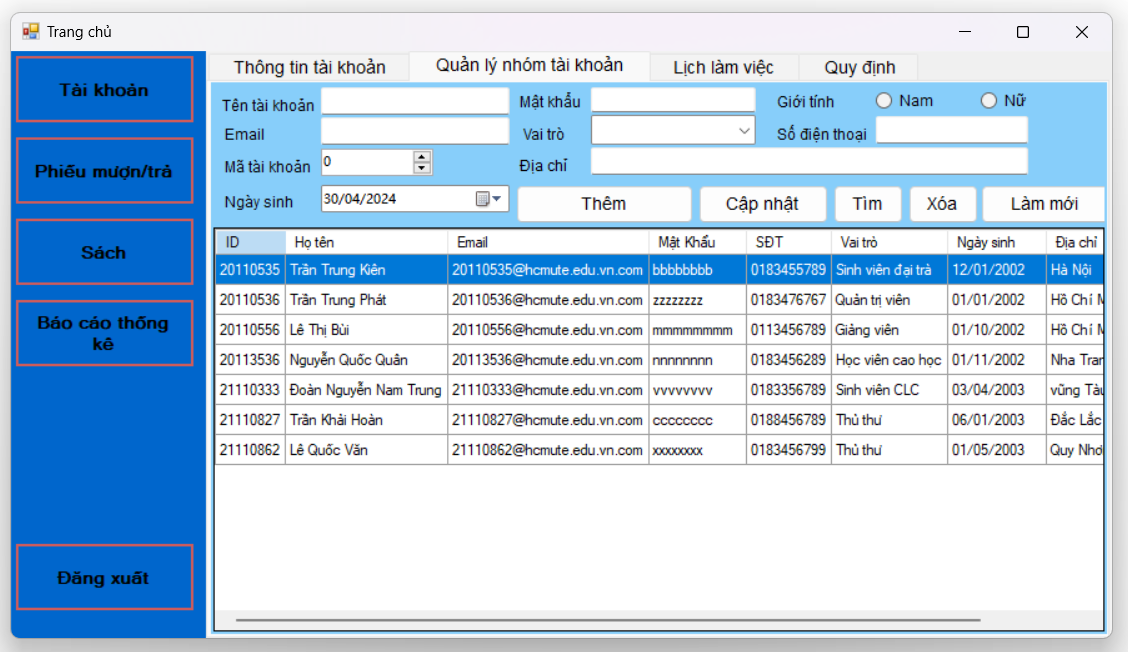
| public DataTable LoadReaderPenalty(int maTaiKhoan, int type)  {  string query = "SELECT \* FROM FN\_Reader\_Penalty\_List( @MaTaiKhoan , @type )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { maTaiKhoan, type });  return data;  } |
| --- |

### 

### **5.2.6. Form Quy định mượn sách thư viện**



### **5.2.7. Form Quản lý nhóm tài khoản**

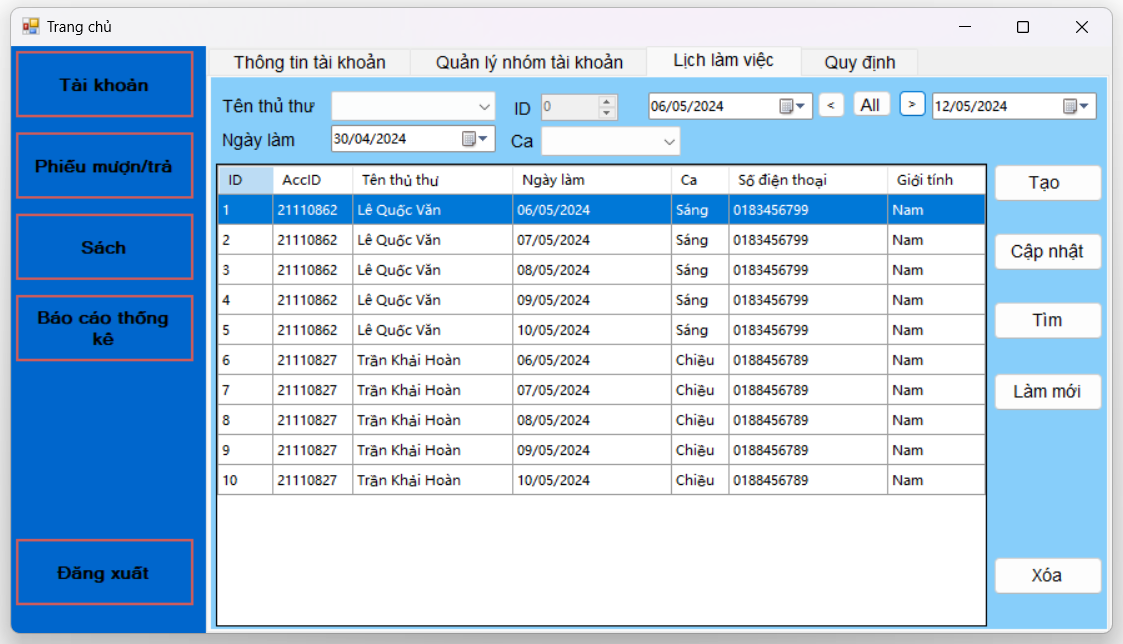


| CREATE VIEW VW\_Account\_List AS  SELECT  \*  FROM dbo.TaiKhoan |
| --- |

Thực thi trên C#:

| public DataTable LoadAccountList()  {  string query = "SELECT \* FROM VW\_Account\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### **5.2.8. Form Quản lý lịch làm việc**

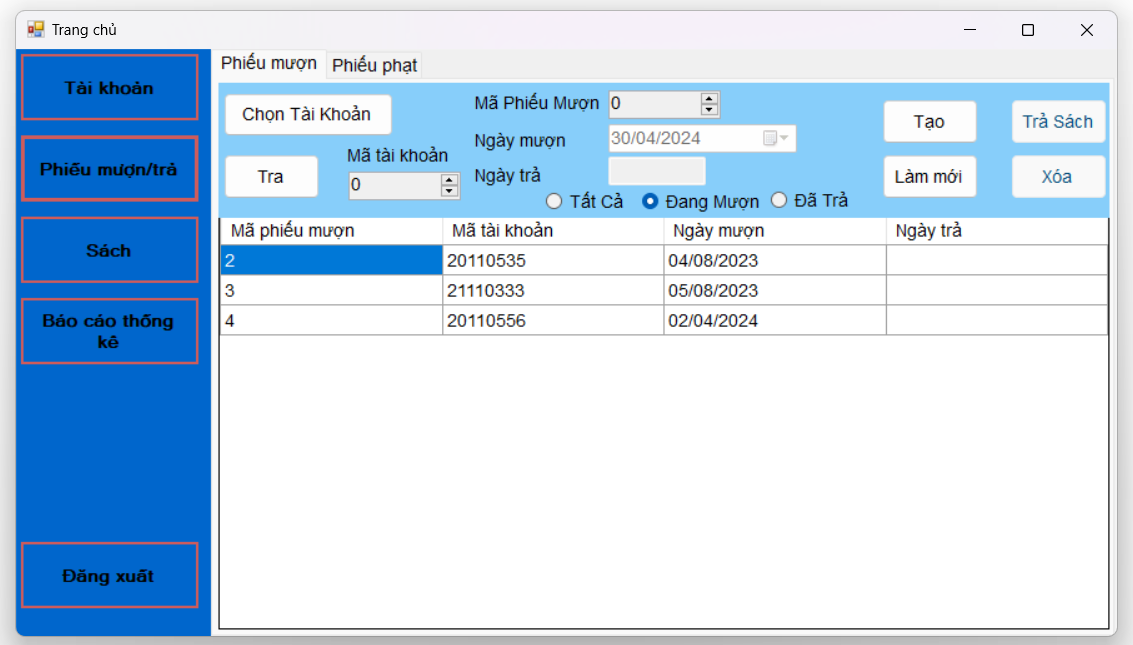


| CREATE OR ALTER PROCEDURE SP\_Get\_Schedule  (  @NgayDauTuan date = NULL,  @NgayCuoiTuan date = NULL,  @MaTaiKhoan int = NULL  )  AS  BEGIN  SELECT \* FROM VW\_Schedule\_List  WHERE  (@NgayDauTuan IS NULL OR NgayLam >= @NgayDauTuan) AND  (@NgayCuoiTuan IS NULL OR NgayLam <= @NgayCuoiTuan) AND  (@MaTaiKhoan IS NULL OR MaTaiKhoan = @MaTaiKhoan)  END; |
| --- |

Thực thi trên C#:

| public DataTable LoadLichLamViecList(DateTime weekstart, DateTime weekend, string maTaiKhoan = null)  {  if (maTaiKhoan == null)  {  string query = "SP\_Get\_Schedule @NgayDauTuan , @NgayCuoiTuan , @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { weekstart, weekend, DBNull.Value });  return data;  }  else  {  int id = Convert.ToInt32(maTaiKhoan);  string query = "SP\_Get\_Schedule @NgayDauTuan , @NgayCuoiTuan , @MaTaiKhoan ";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { weekstart, weekend, id });  return data;  }  } |
| --- |

### **5.2.9. Form Quản lý phiếu mượn**

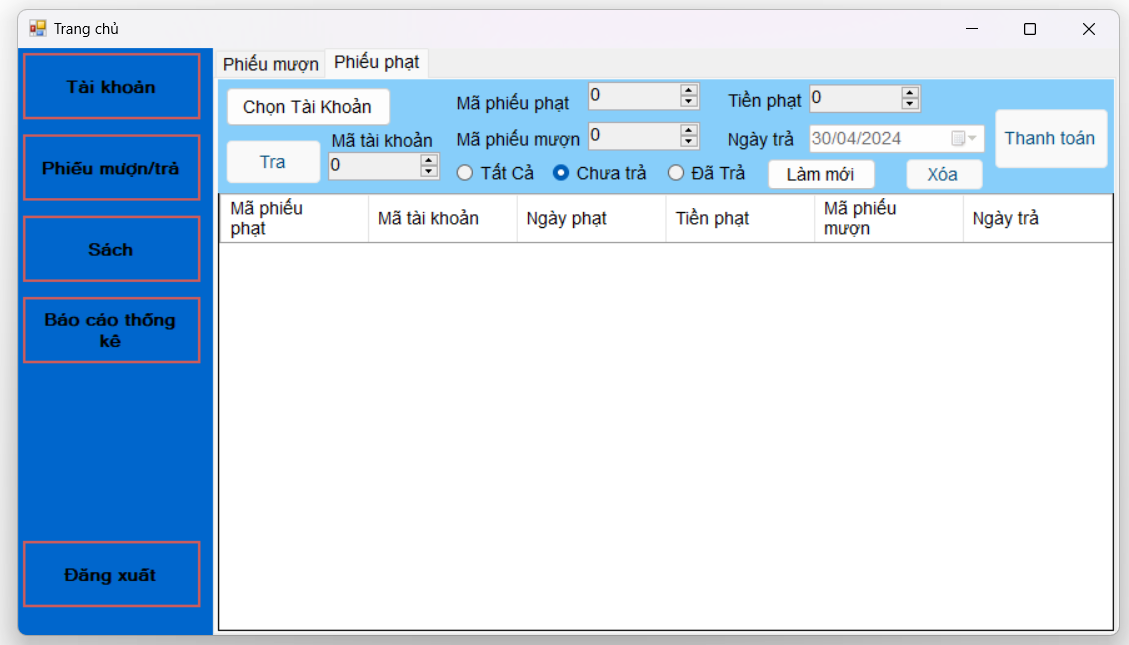


| Create VIEW VW\_Book\_Loan\_Coupon\_List AS  SELECT  \*  FROM dbo.PhieuMuonSach |
| --- |

Thực thi trên C#:

| public DataTable LoadBookLoanCouponList()  {  string query = "SELECT \* FROM VW\_Book\_Loan\_Coupon\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### 5.2.10. Form Quản lý phiếu phạt

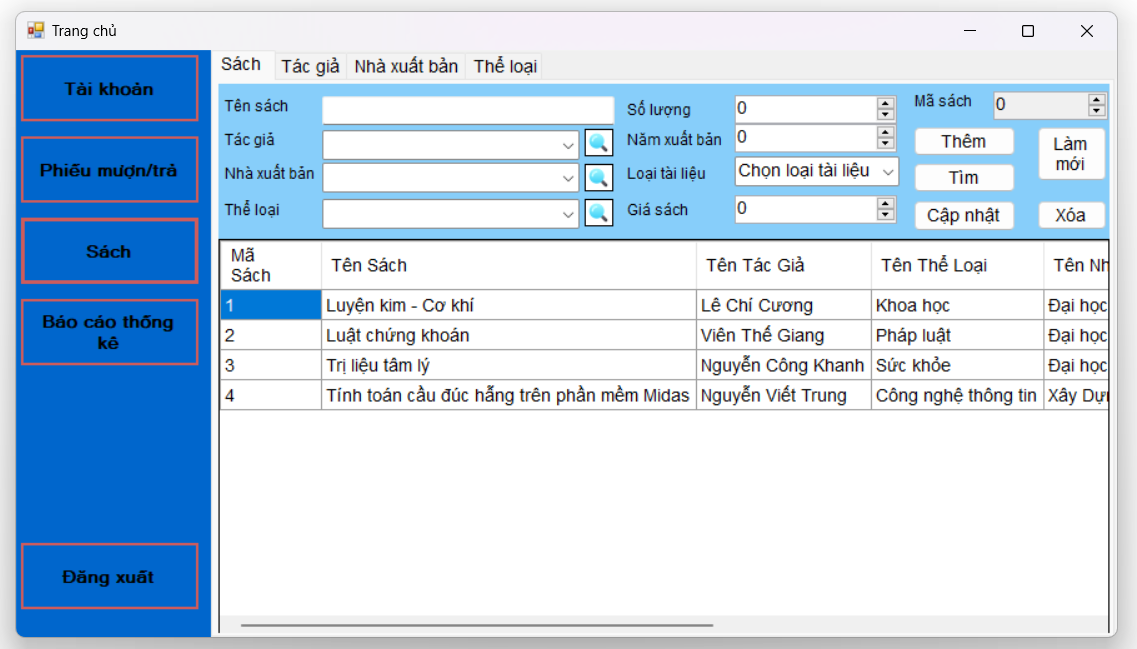


| CREATE VIEW VW\_PhieuPhat\_List AS  SELECT PP.MaPhieuPhat, PP.MaPhieuMuon,PP.NgayTra,PMS.MaTaiKhoan,PMS.NgayTra AS NgayTraSach  FROM dbo.PhieuPhat PP  JOIN dbo.PhieuMuonSach PMS ON PMS.MaPhieuMuon= PP.MaPhieuMuon |
| --- |

Thực thi trên C#:

| public DataTable LoadPhieuPhatList()  {  string query = "SELECT \* FROM VW\_PhieuPhat\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### **5.2.11. Form Quản lý sách**

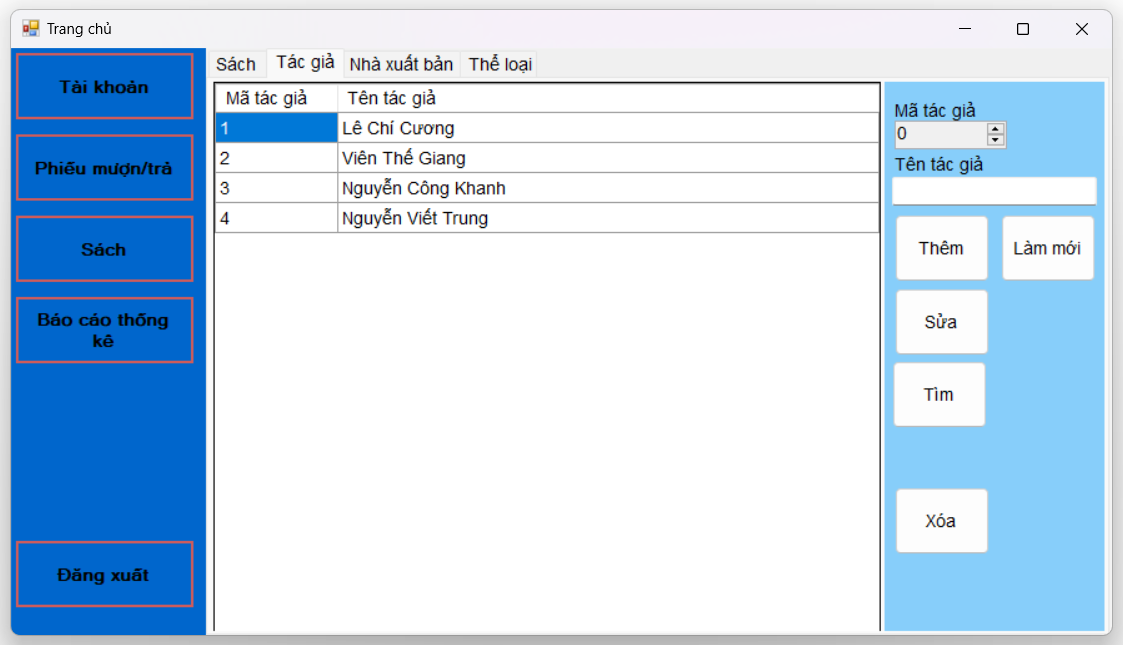


| CREATE VIEW VW\_Book\_List AS  SELECT  S.MaSach,  TG.TenTacGia,  TL.TenTheLoai,  NXB.TenNhaXuatBan,  S.TenSach,  S.LoaiTaiLieu,  S.NamXuatBan,  S.GiaSach,  S.SoLuong  FROM  Sach S  INNER JOIN  TacGia TG ON S.MaTacGia = TG.MaTacGia  INNER JOIN  TheLoai TL ON S.MaTheLoai = TL.MaTheLoai  INNER JOIN  NhaXuatBan NXB ON S.MaNhaXuatBan = NXB.MaNhaXuatBan; |
| --- |

Thực thi trên C#:

| public DataTable LoadBookList()  {  string query = "SELECT \* FROM VW\_Book\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### **5.2.12. Form Quản lý tác giả**

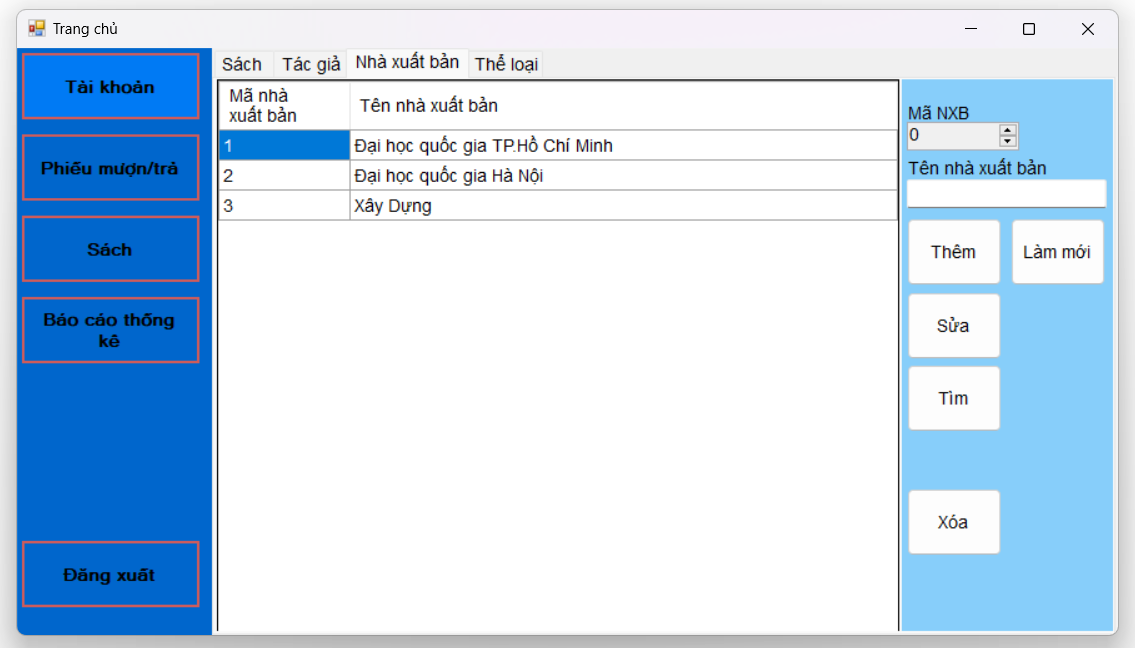


| CREATE VIEW VW\_TacGia\_List AS  SELECT \*  FROM dbo.TacGia |
| --- |

Thực thi trên C#:

| public DataTable LoadTacGiaList()  {  string query = "Select \* from VW\_TacGia\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### **5.2.13. Form Quản lý nhà xuất bản**

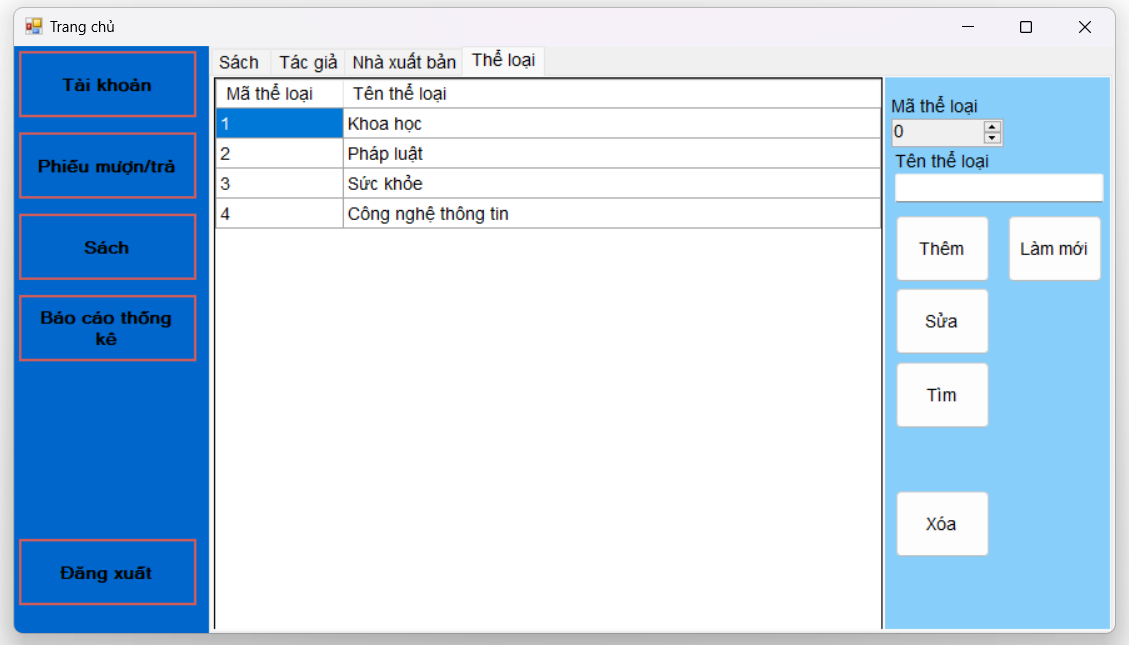


| Create VIEW VW\_NhaXuatBan\_List AS  SELECT  MaNhaXuatBan,  TenNhaXuatBan  FROM dbo.NhaXuatBan |
| --- |

Thực thi trên C#:

| public DataTable LoadNXBList()  {  string query = "SELECT \* FROM VW\_NhaXuatBan\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

### **5.2.14. Form Quản lý thể loại**

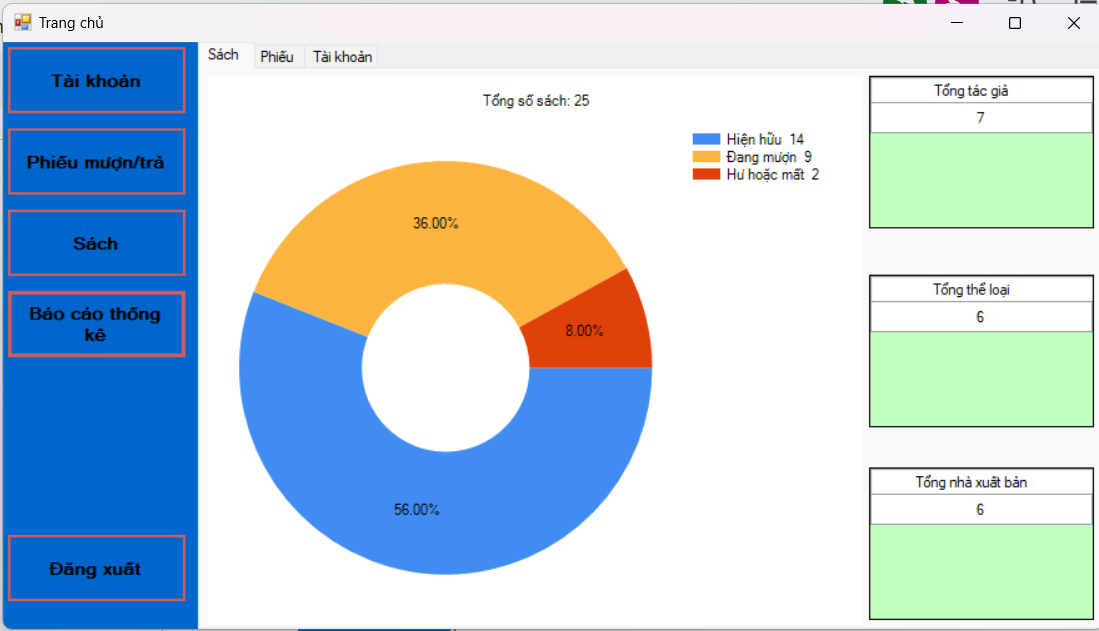


| CREATE VIEW VW\_TheLoai\_List AS  SELECT \*  FROM dbo.TheLoai |
| --- |

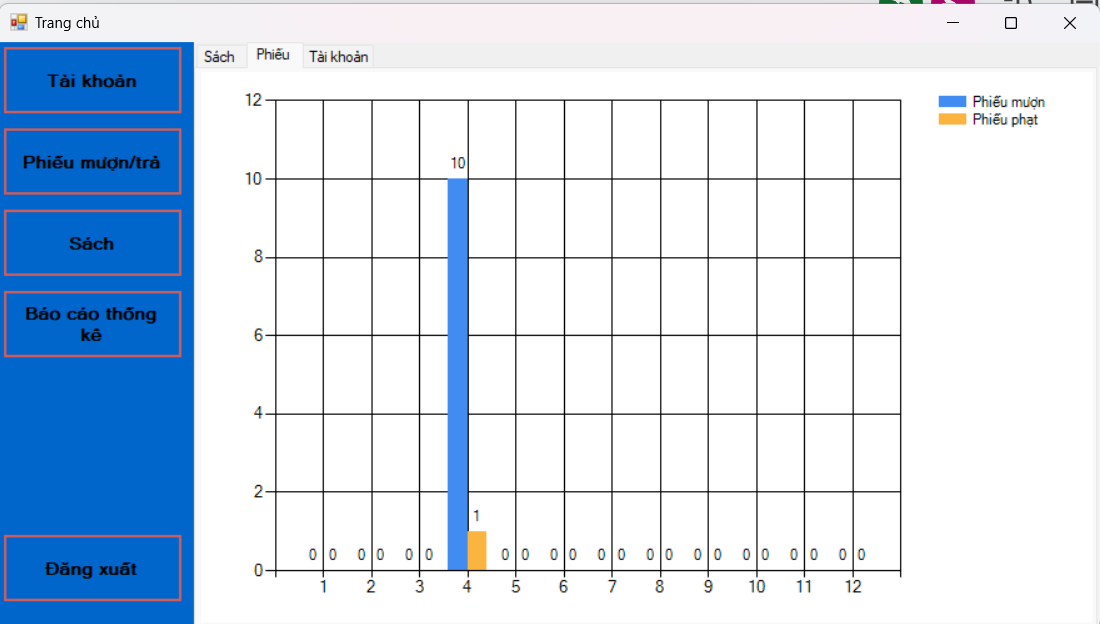
Thực thi trên C#:

| public DataTable LoadTheLoaiList()  {  string query = "SELECT \* FROM VW\_TheLoai\_List";  DataTable data = DataProvider.Instance.ExecuteQuery(query);  return data;  } |
| --- |

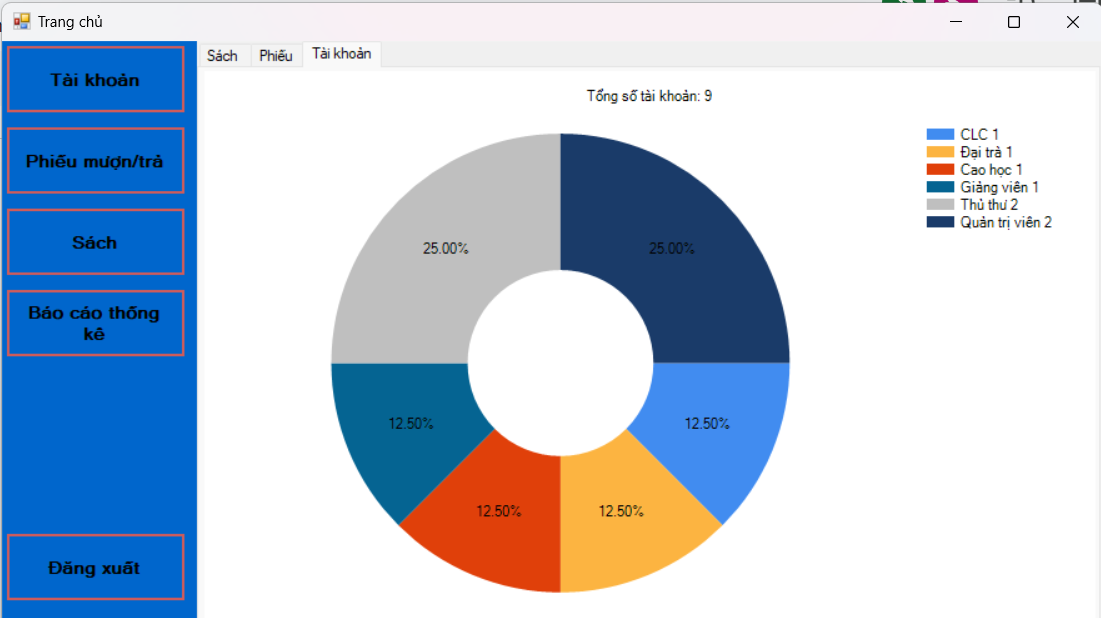
### **5.2.15. Form Thống kê sách**



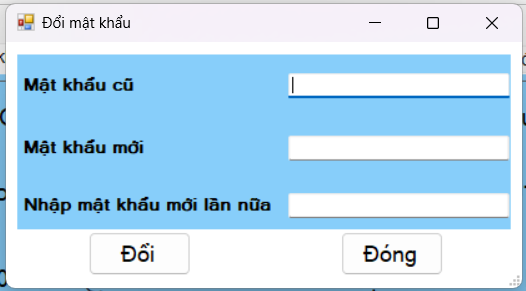
### **5.2.16. Form Thống kê phiếu mượn và phạt**



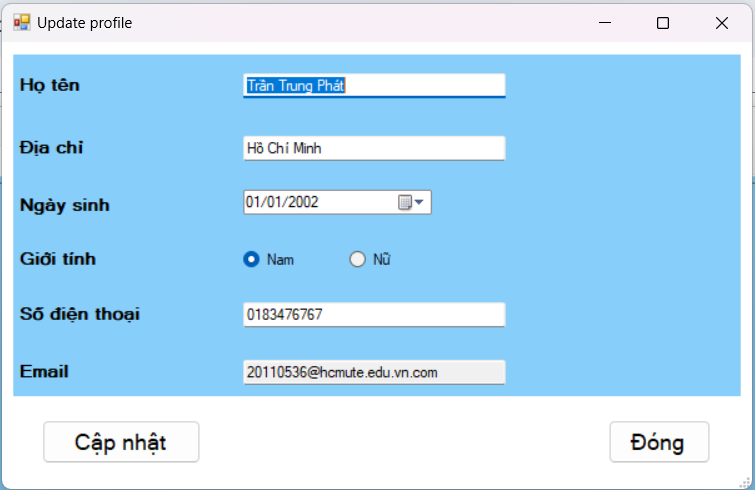
### **5.2.17. Form Thống kê trên vai trò tài khoản**



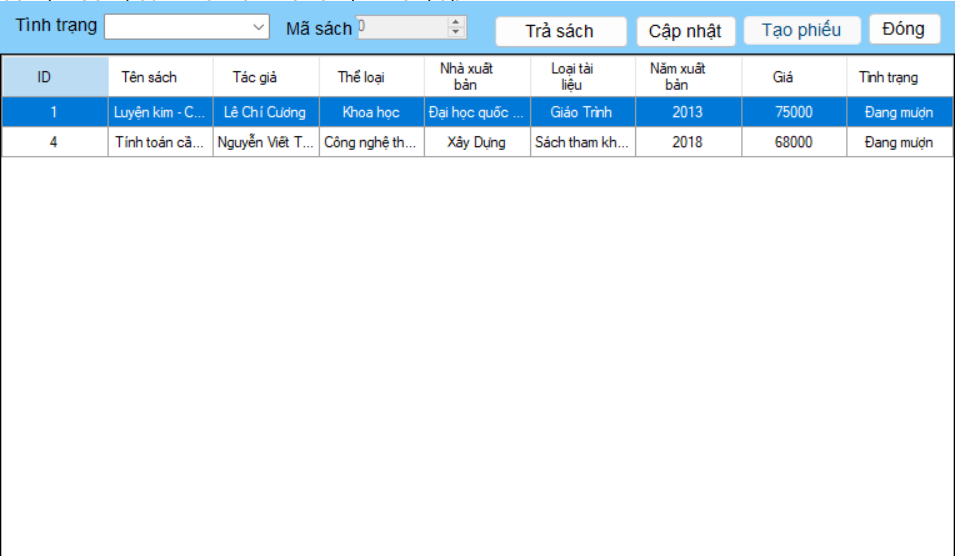
### **5.2.18. Form Đổi mật khẩu tài khoản**



### **5.2.19. Form Cập nhật tài khoản**



### **5.2.20. Form Xem chi tiết sách mượn**



| CREATE OR ALTER FUNCTION FN\_Reader\_Borrowed\_Detail  (  @MaPhieuMuon int = NULL,  @MaPhieuPhat int = NULL  )  RETURNS TABLE  AS  RETURN (  SELECT  CS.MaSach AS N'ID',  TenSach AS N'Tên sách',  TenTacGia AS N'Tác giả',  TenTheLoai AS N'Thể loại',  TenNhaXuatBan AS N'Nhà xuất bản',  LoaiTaiLieu AS N'Loại tài liệu',  NamXuatBan AS N Năm xuất bản',  S.GiaSach AS N'Giá',  CS.TinhTrang AS N'Tình trạng'  FROM dbo.CuonSach CS  JOIN dbo.VW\_Book\_List S ON S.MaSach = CS.MaSach  WHERE  (@MaPhieuMuon IS NOT NULL AND CS.MaPhieuMuon = @MaPhieuMuon) OR  (@MaPhieuPhat IS NOT NULL AND CS.MaPhieuMuon =  (SELECT MaPhieuMuon FROM dbo.PhieuPhat WHERE MaPhieuPhat = @MaPhieuPhat))  ); |
| --- |

Thực thi trên C#

| public DataTable LoadBookListByBorrowedOrPenaltyID(int muon, int phat)  {  if (muon != 0)  {  string query = "SELECT \* FROM dbo.FN\_Reader\_Borrowed\_Detail( @MaPhieuMuon , @MaPhieuPhat )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { muon, DBNull.Value });  return data;  }  else  {  string query = "SELECT \* FROM dbo.FN\_Reader\_Borrowed\_Detail( @MaPhieuMuon , @MaPhieuPhat )";  DataTable data = DataProvider.Instance.ExecuteQuery(query, new object[] { DBNull.Value, phat });  return data;  }  } |
| --- |