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

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

-----🙞🙞🙞🙞🙞-----



**ĐỀ TÀI CUỐI KÌ MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**TÊN ĐỀ TÀI:**

**QUẢN LÝ THƯ VIỆN TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**GVHD:** TS. Nguyễn Thành Sơn

**Lớp HP:** DBMS330284\_22\_2\_02CLC

**Nhóm thực hiện:** Nhóm 03

**Học kì:** II

**Năm học:** 2022-2023

*Thành phố Hồ Chí Minh, tháng 2 năm 2023*

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ II NĂM HỌC 2022-2023**

**Nhóm 03**

***Đề tài: Quản lý thư viện***

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên** |
| 20110127 | Nguyễn Tiến Dũng |
| 21110816 | Nguyễn Thành Châu |
| 21110311 | Nguyễn Trường Thịnh |
| 21110312 | Ngô Ngọc Thông |

**NHẬN XÉT CỦA GIẢNG VIÊN**

Tp. Hồ Chí Minh, ngày … tháng … năm 2023

Giảng viên chấm điểm

**MỤC LỤC**

[**CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG 1**](#_Toc134478120)

[**1. Đặc tả đề tài 1**](#_Toc134478121)

[**1.1 Vấn đề thực tế 1**](#_Toc134478122)

[**1.2 Phạm vi đề tài 1**](#_Toc134478123)

[**1.3 Mô tả bài toán 1**](#_Toc134478124)

[**1.4 Mô tả chức năng của bài toán 2**](#_Toc134478125)

[**1.5 Chính sách về mượn trả 2**](#_Toc134478126)

[**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 4**](#_Toc134478127)

[**1. Thiết kế cơ sở dữ liệu mức quan niệm 4**](#_Toc134478128)

[**2. Thiết kế cơ sở dữ liệu mức logic 4**](#_Toc134478129)

[**3. Các ràng buộc 5**](#_Toc134478130)

[**4. Cài đặt cơ sở dữ liệu và các ràng buộc 6**](#_Toc134478131)

[**4.1 Tạo bảng và các ràng buộc đơn giản 6**](#_Toc134478132)

[**4.2 Các ràng buộc khác 8**](#_Toc134478133)

[**4.3 Trigger 9**](#_Toc134478134)

[**4.4 View 13**](#_Toc134478135)

[**CHƯƠNG 3. THIẾT KẾ CÁC CHỨC NĂNG 16**](#_Toc134478136)

[**0. Kết nối cơ sở dữ liệu 16**](#_Toc134478137)

[**1. Procedure 16**](#_Toc134478138)

[**1.1 Quản lý Sách 16**](#_Toc134478139)

[**1.2 Quản lý độc giả 20**](#_Toc134478140)

[**1.3 Quản lý Mượn trả 23**](#_Toc134478141)

[**1.4 Quản lý Tác giả 27**](#_Toc134478142)

[**1.5 Quản lý Nhà xuất bản 28**](#_Toc134478143)

[**1.6 Quản lý Chuyên ngành 29**](#_Toc134478144)

[**1.7 Quản lý Nhân viên 30**](#_Toc134478145)

[**1.8 Quản lý đăng nhập 34**](#_Toc134478146)

[**2. Function 35**](#_Toc134478147)

[**2.1 Tự động tạo mã 35**](#_Toc134478148)

[**2.2 Tìm kiếm 37**](#_Toc134478149)

[**2.3 Lấy mã nhân viên khi đăng nhập 40**](#_Toc134478150)

[**2.4 Tính tiền phạt 40**](#_Toc134478151)

[**CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN 42**](#_Toc134478152)

[**1. Nhân viên 42**](#_Toc134478153)

[**2. Quản trị viên 42**](#_Toc134478154)

[**3. Trigger tạo User 42**](#_Toc134478155)

[**CHƯƠNG 5. THIẾT KẾ GIAO DIỆN HỆ THỐNG 44**](#_Toc134478156)

[**1. Đăng nhập 44**](#_Toc134478157)

[**2. Độc giả 44**](#_Toc134478158)

[**2.1 Form độc giả 44**](#_Toc134478159)

[**2.2 Dialog Thêm độc giả 45**](#_Toc134478160)

[**3. Sách 46**](#_Toc134478161)

[**3.1 Form Sách 46**](#_Toc134478162)

[**3.2 Dialog Thêm sách 47**](#_Toc134478163)

[**4. Tác giả 48**](#_Toc134478164)

[**5. Nhà xuất bản 49**](#_Toc134478165)

[**6. Chuyên ngành 50**](#_Toc134478166)

[**7. Mượn trả sách 51**](#_Toc134478167)

[**7.1 Form Mượn trả sách 51**](#_Toc134478168)

[**7.2 Dialog cho mượn sách 52**](#_Toc134478169)

[**7.3 Dialog Trả sách 52**](#_Toc134478170)

[**8. Nhân viên 53**](#_Toc134478171)

[**8.1 Form Nhân viên 53**](#_Toc134478172)

[**8.2 Dialog Thêm nhân viên 54**](#_Toc134478173)

[**9. Đổi mật khẩu 55**](#_Toc134478174)

# CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG

## Đặc tả đề tài

### Vấn đề thực tế

Đọc sách là một nhu cầu cơ bản trong quá trình học tập, hay cả trong đời sống sinh hoạt hàng ngày của con người. Tuy nhiên, việc mua và sở hữu một quyển sách mới không phải luôn là nhu cầu thiết yếu của người đọc. Vậy nên mới có sự xuất hiện của thư viện trong cuộc sống ngày nay để phục vụ những người có nhu cầu đọc nhưng không có nhu cầu sở hữu cuốn sách đó. Việc có một ứng dụng phần mềm để quản lý sách cũng như việc mượn và trả sách của độc giả là một công cụ hỗ trợ rất hữu ích và cần thiết đối với việc quản lý thư viện.

### Phạm vi đề tài

Do phần mềm chỉ phục vụ mục đích học tập nên không hoàn toàn có thể thay thế các phần mềm quản lý thư viện khác ở trên thực tế. Tuy nhiên, phần mềm vẫn có thể mô phỏng cách hoạt động của ứng dụng ngoài thực tế một cách đơn giản.

### Mô tả bài toán

Xây dựng hệ thống quản lý thư viện cho trường ĐH Sư phạm Kỹ thuật Tp.HCM gồm các hoạt động chính là: quản lý thông tin sách, quản lý thông tin độc giả, quản lý hoạt động mượn trả sách. Thông tin cụ thể cho các hoạt động trên:

Khi lần đầu mượn sách ở thư viện, độc giả cung cấp thẻ sinh viên/cán bộ cho thư viện, thư viện sẽ làm thẻ (không phải thẻ vật lý, mà chỉ lưu trên cơ sở dữ liệu), có thời hạn 1 năm. Thông tin *Độc Giả* lưu trong CSDL bao gồm: *Mã độc giả, Đối tượng, Họ tên, Giới tính, Ngày sinh, SĐT, Email, Ngày làm thẻ, Ngày hết hạn.*

*Sách* được lưu trong CSDL gồm các thuộc tính: *Mã sách, Tên sách, Loại sách, Nhà xuất bản, Chuyên ngành, Giá bìa, Số lượng*. *Nhà xuất bản, Chuyên ngành* có bảng riêng gồm *Mã* và *Tên* để những giá trị này trong bảng Sách tham chiếu đến. Riêng đối với *Tác Giả*, do Sách – Tác giả là quan hệ nhiều – nhiều nên cần một bảng trung gian là *Sách Tác Giả* để lưu trữ bao gồm: *Mã tác giả, Mã sách*

Về mượn sách, *Phiếu Mượn* lưu những thông tin tổng quan về lần mượn sách bao gồm: *Mã phiếu mượn, Người mượn, Nhân viên cho mượn, Ngày mượn, Số lượng*

Chi tiết về từng sách được mượn lưu trong bảng *Chi Tiết Mượn Trả*, bảng này cũng lưu những thông tin khi sách được trả lại, bao gồm: *Mã phiếu mượn, Mã sách, Ngày hết hạn, Ngày trả, Nhân viên trả, Tình trạng khi trả, Tiền phạt hư hỏng, Tiền phạt trễ hạn.*

Về nhân viên, nhân viên sẽ được lưu trong bảng *Nhân Viên* bao gồm: *Mã nhân viên, Họ tên, Giới tính, SĐT, Email, Ngày vào làm, Tình trạng làm việc*. Mỗi nhân viên có một tài khoản đăng nhập được lưu trong bảng *Đăng nhập* bao gồm: *Mã nhân viên, Tên đăng nhập, Mật khẩu, Loại tài khoản.*

### Mô tả chức năng của bài toán

*Đối với tài khoản Thủ thư:*

Quản lý sách: thêm, xoá sách, chỉnh sửa thông tin liên quan đến sách. Tìm kiếm sách theo tên hoặc mã sách. Xem các thống kê về sách.

Quản lý độc giả: thêm, xoá, chỉnh sửa các thông tin của độc giả. Tìm kiếm độc giả theo mã độc giả.

Quản lý mượn trả: quản lý các hoạt động mượn và trả sách thông qua việc thêm, xoá, chỉnh sửa Phiếu mượn và Chi tiết mượn trả. Tìm kiếm thông tin sách được mượn trên hệ thống thông qua mã phiếu mượn. Xem các thống kê về việc mượn và trả sách

Chỉnh sửa thông tin cá nhân của mình trong bảng nhân viên

*Đối với tài khoản Quản trị viên:*

Thực hiện được tất cả các quyền của nhân viên

Quản lý nhân viên: quản lý về nhân viên, như việc thêm, xoá, chỉnh sửa các thông tin của nhân viên

### Chính sách về mượn trả

Chính sách về mượn trả chúng ta cần làm rõ trong hệ thống là chính sách về thời gian, số lượng sách được mượn và các quy định phạt

*Về thời gian và số lượng sách được mượn:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đối tượng độc giả** | **Sách tham khảo** | | **Giáo trình** | |
| Số lượng | Thời gian | Số lượng | Thời gian |
| Sinh viên đại trà | 10 | 4 tuần | 15 | 1 học kỳ |
| Sinh viên khoa ĐT CLC | 10 | 4 tuần | 20 | 1 học kỳ |
| Sinh viên khoa ĐT Quốc tế | 10 | 4 tuần | 15 | 1 học kỳ |
| Học viên cao học | 5 | 4 tuần | 5 | 8 tuần |
| Cán bộ viên chức | 10 | 1 năm | 5 | 1 năm |
| Ngoài trường | Chỉ đọc tại chỗ, không phục vụ mượn về | | | |

*Về quy định phạt:*

Khi trả sách trễ so với ngày hẹn người mượn phải chịu mức phạt 1000đ/ngày cho mỗi sách

Nếu làm mất tài liệu, phải đền đúng tài liệu làm mất + 10.000đ phí xử lý nghiệp vụ

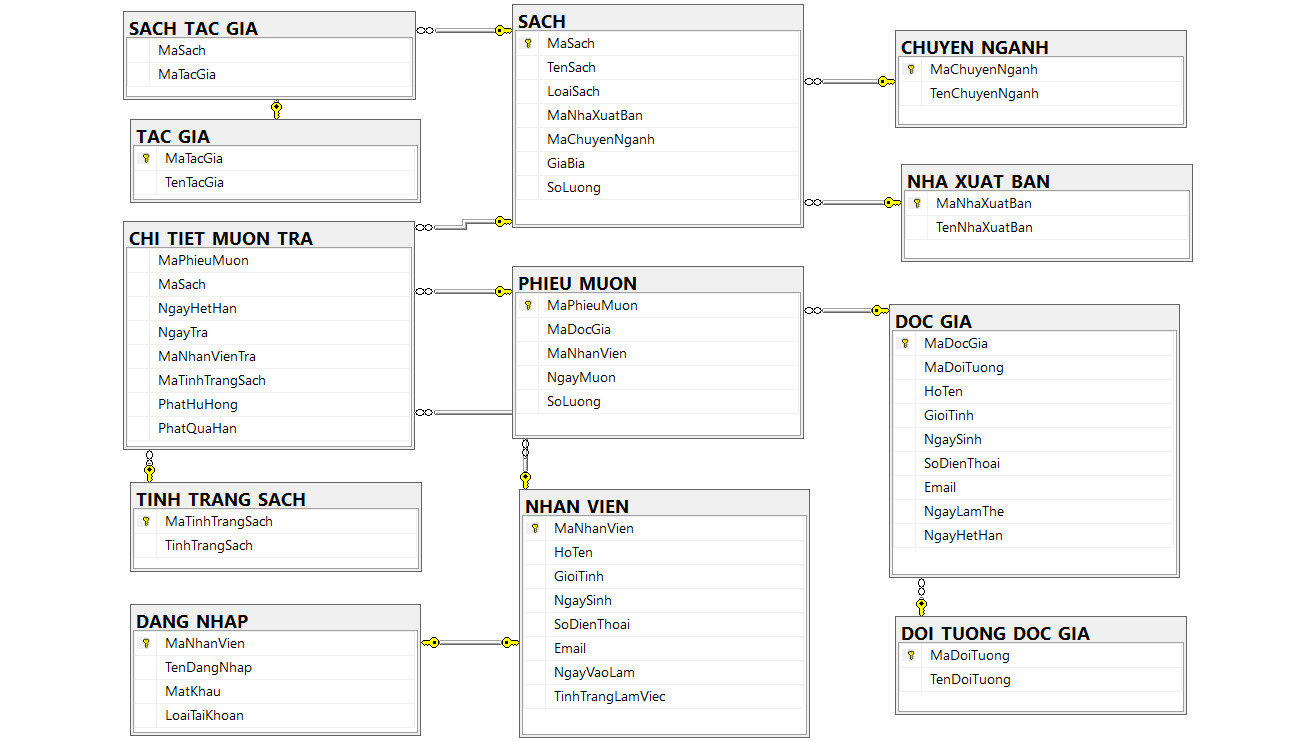
Nếu làm mất nhưng không thể đền đúng tài liệu thì phải bồi thường gấp 5 lần giá trị tài liệu gốc + 10.000đ phí xử lý nghiệp vụ

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Thiết kế cơ sở dữ liệu mức quan niệm

## Thiết kế cơ sở dữ liệu mức logic

SACH(MaSach, TenSach, LoaiSach, MaNhaXuatBan, MaChuyenNganh, GiaBia, SoLuong)  
NHA\_XUAT\_BAN(MaNhaXuatBan, TenNhaXuatBan)  
CHUYEN\_NGANH(MaChuyenNganh, TenChuyenNganh)  
TAC\_GIA(MaTacGia, TenTacGia)  
SACH\_TAC\_GIA(MaTacGia, MaSach)  
DOC\_GIA(MaDocGia, MaDoiTuong, HoTen, GioiTinh, NgaySinh, SoDienThoai, Email, NgayLamThe, NgayHetHan)  
DOI\_TUONG\_DOC\_GIA(MaDoiTuong, TenDoiTuong)  
NHAN\_VIEN(MaNhanVien, HoTen, GioiTinh, NgaySinh, SoDienThoai, Email, NgayVaoLam, TinhTrangLamViec)  
DANG\_NHAP(MaNhanVien, TenDangNhap, MatKhau, LoaiTaiKhoan)  
PHIEU\_MUON(MaPhieuMuon, MaDocGia, MaNhanVien, NgayMuon, SoLuong)  
CHI\_TIET\_MUON\_TRA(MaPhieuMuon, MaSach, NgayHetHan, NgayTra, MaNhanVienTra, MaTinhTrangSach, PhatHuHong, PhatQuaHan)  
TINH\_TRANG\_SACH(MaTinhTrangSach, TinhTrangSach)



## Các ràng buộc

|  |  |
| --- | --- |
| **Bảng** | **Ràng buộc** |
| SACH | - Ràng buộc khoá chính MaSach  - Ràng buộc khoá ngoại MaNhaXuatBan tham chiếu đến bảng **NHA\_XUAT\_BAN**  - Ràng buộc khoá ngoại MaChuyenNganh tham chiếu đến bảng **CHUYEN\_NGANH** |
| NHA\_XUAT\_BAN | - Ràng buộc khoá chính MaNhaXuatBan |
| CHUYEN\_NGANH | - Ràng buộc khoá chính MaChuyenNganh |
| TAC\_GIA | - Ràng buộc khoá chính MaTacGia |
| SACH\_TAC\_GIA | - Ràng buộc khoá ngoại MaTacGia tham chiếu đến bảng **TAC\_GIA**  **-** Ràng buộc khoá ngoại MaSach tham chiếu đến bảng **SACH** |
| DOC\_GIA | - Ràng buộc khoá chính MaDocGia  - Ràng buộc khoá ngoại MaDoiTuong tham chiếu đến bảng **DOI\_TUONG\_DOC\_GIA** |
| DOI\_TUONG\_DOC\_GIA | - Ràng buộc khoá chính MaDoiTuong |
| NHAN\_VIEN | - Ràng buộc khoá chính MaNhanVien |
| DANG\_NHAP | - Ràng buộc khoá chính MaNhanVien  - Ràng buộc khoá ngoại MaNhanVien tham chiếu đến bảng **NHAN\_VIEN** |
| PHIEU\_MUON | - Ràng buộc khoá chính MaPhieuMuon  - Ràng buộc khoá ngoại MaDocGia tham chiếu đến bảng **DOC\_GIA**  - Ràng buộc khoá ngoại MaNhanVien tham chiếu đến bảng **NHAN\_VIEN** |
| CHI\_TIET\_MUON\_TRA | - Ràng buộc khoá ngoại MaPhieuMuon tham chiếu đến bảng **PHIEU\_MUON**  - Ràng buộc khoá ngoại MaSach tham chiếu đến bảng **SACH**  - Ràng buộc khoá ngoại MaNhanVienTra tham chiếu đến bảng **NHAN\_VIEN** |
| TINH\_TRANG\_SACH | - Ràng buộc khoá chính MaTinhTrangSach |

## Cài đặt cơ sở dữ liệu và các ràng buộc

### 4.1 Tạo bảng và các ràng buộc đơn giản

**-- 4.1.1 Bảng Nhà xuất bản**

CREATE TABLE NHA\_XUAT\_BAN (

MaNhaXuatBan varchar(10) primary key,

TenNhaXuatBan nvarchar(50) not null,

);

**-- 4.1.2 Bảng Tác giả**

CREATE TABLE TAC\_GIA (

MaTacGia varchar(10) primary key,

TenTacGia nvarchar(50) not null,

);

**-- 4.1.3 Bảng Chuyên ngành**

CREATE TABLE CHUYEN\_NGANH (

MaChuyenNganh varchar(10) primary key,

TenChuyenNganh nvarchar(50) not null,

);

**-- 4.1.4 Bảng Sách**

CREATE TABLE SACH (

MaSach varchar(10) primary key,

TenSach nvarchar(100) not null,

LoaiSach bit not null, -- 0: giáo trình -- 1: sách tham khảo

MaNhaXuatBan varchar(10) not null,

MaChuyenNganh varchar(10) not null,

GiaBia decimal(9,3) not null,

SoLuong int not null,

);

**-- 4.1.5 Bảng Sách Tác Giả**

CREATE TABLE SACH\_TAC\_GIA (

MaSach varchar(10) not null,

MaTacGia varchar(10) not null,

);

**-- 4.1.6 Bảng Đối tượng độc giả**

CREATE TABLE DOI\_TUONG\_DOC\_GIA (

MaDoiTuong int identity(1,1) primary key,

TenDoiTuong nvarchar (50) not null,

);

**-- 4.1.7 Bảng Độc giả**

CREATE TABLE DOC\_GIA (

MaDocGia varchar(10) primary key,

MaDoiTuong int not null,

HoTen nvarchar(50) not null,

GioiTinh bit not null,

NgaySinh date not null,

SoDienThoai varchar(10),

Email varchar(50),

NgayLamThe date not null,

NgayHetHan date not null,

);

**-- 4.1.8 Bảng Nhân viên**

CREATE TABLE NHAN\_VIEN (

MaNhanVien varchar(10) primary key,

HoTen nvarchar(50) not null,

GioiTinh bit not null,

NgaySinh date,

SoDienThoai varchar(10),

Email varchar(50),

NgayVaoLam date not null,

TinhTrangLamViec bit default 1 , -- 1: còn làm việc -- 0: nghỉ việc

);

**-- 4.1.9 Bảng Đăng nhập**

CREATE TABLE DANG\_NHAP (

MaNhanVien varchar(10) primary key,

TenDangNhap varchar(20) unique not null,

MatKhau varchar(20) not null,

LoaiTaiKhoan bit not null, -- 0: quản trị viên -- 1: thủ thư

);

**-- 4.1.10 Bảng Phiếu mượn**

CREATE TABLE PHIEU\_MUON (

MaPhieuMuon varchar(10) primary key,

MaDocGia varchar(10) not null,

MaNhanVien varchar(10) not null,

NgayMuon date not null,

SoLuong int not null default 0,

);

**-- 4.1.11 Bảng Tình trạng sách**

CREATE TABLE TINH\_TRANG\_SACH (

MaTinhTrangSach smallint identity(1,1) primary key,

TinhTrangSach nvarchar(50) not null,

);

**-- 4.1.12 Bảng chi tiết mượn trả**

CREATE TABLE CHI\_TIET\_MUON\_TRA (

MaPhieuMuon varchar(10) not null,

MaSach varchar(10) not null,

NgayHetHan date not null,

NgayTra date,

MaNhanVienTra varchar(10),

MaTinhTrangSach smallint,

PhatHuHong decimal(9,3),

PhatQuaHan decimal(9,3),

);

### 4.2 Các ràng buộc khác

**-- 4.2.1 MẶC ĐỊNH THÊM MÃ TÁC GIẢ THEO FUNCTION AUTO\_MATACGIA()**

ALTER TABLE TAC\_GIA

ADD CONSTRAINT default\_matacgia DEFAULT dbo.Auto\_MatacGia() FOR MaTacGia

**-- 4.2.2 MẶC ĐỊNH THÊM MÃ NHÀ XUẤT BẢN THEO FUNCTION AUTO\_MANHAXUATBAN()**

ALTER TABLE NHA\_XUAT\_BAN

ADD CONSTRAINT default\_manhaxuatban DEFAULT dbo.Auto\_MaNhaXuatBan() FOR MaNhaXuatBan

**-- 4.2.3 MẶC ĐỊNH THÊM MÃ CHUYÊN NGÀNH THEO FUNCTION AUTO\_MACHUYENNGANH()**

ALTER TABLE CHUYEN\_NGANH

ADD CONSTRAINT default\_machuyennganh DEFAULT dbo.Auto\_MaChuyenNganh() FOR MaChuyenNganh

**-- 4.2.4 CÁC RÀNG BUỘC KHOÁ NGOẠI VÀ MẶC ĐỊNH THÊM MÃ SÁCH THEO FUNCTION AUTO\_MASACH()**

ALTER TABLE SACH

ADD CONSTRAINT fk\_manhaxuatban\_sach\_nhaxuatban FOREIGN KEY (MaNhaXuatBan) REFERENCES NHA\_XUAT\_BAN(MaNhaXuatBan) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT fk\_machuyennganh\_sach\_chuyennganh FOREIGN KEY (MaChuyenNganh) REFERENCES CHUYEN\_NGANH(MaChuyenNganh) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT default\_masach DEFAULT dbo.Auto\_MaSach() FOR MaSach

**-- 4.2.5 CÁC RÀNG BUỘC KHOÁ NGOẠI**

ALTER TABLE SACH\_TAC\_GIA

ADD CONSTRAINT fk\_matacgia\_sachtacgia\_tacgia FOREIGN KEY (MaTacGia) REFERENCES TAC\_GIA(MaTacGia) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_masach\_sachtacgia\_sach FOREIGN KEY (MaSach) REFERENCES SACH(MaSach) ON DELETE CASCADE ON UPDATE CASCADE;

**-- 4.2.6 CÁC RÀNG BUỘC KHOÁ NGOẠI**

ALTER TABLE DOC\_GIA

ADD CONSTRAINT fk\_madoituong\_docgia\_doituongdocgia FOREIGN KEY (MaDoiTuong) REFERENCES DOI\_TUONG\_DOC\_GIA(MaDoiTuong) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT check\_ngaysinh\_docgia CHECK (DATEDIFF(year, NgaySinh, GETDATE())>=18),

CONSTRAINT check\_sdt\_docgia CHECK (len(SoDienThoai)=10),

CONSTRAINT check\_email\_docgia CHECK (Email like '%\_\_\_@\_\_\_%.\_\_%');

**-- 4.2.7 CÁC RÀNG BUỘC CHECK ĐỂ KIỂM TRA HỢP LỆ CỦA NGÀY SINH, SĐT, EMAIL KHI THÊM VÀO BẢNG VÀ MẶC ĐỊNH THÊM MÃ NHÂN VIÊN THEO FUNCTION AUTO\_MANHAN VIEN()**

ALTER TABLE NHAN\_VIEN

ADD CONSTRAINT check\_ngaysinh\_nhanvien CHECK (DATEDIFF(year, NgaySinh, GETDATE())>=18),

CONSTRAINT check\_sdt\_nhanvien CHECK (len(SoDienThoai)=10),

CONSTRAINT check\_email\_nhanvien CHECK (Email like '%\_\_\_@\_\_\_%.\_\_%'),

CONSTRAINT default\_manhanvien DEFAULT dbo.Auto\_MaNhanVien() FOR MaNhanVien

**-- 4.2.8 CÁC RÀNG BUỘC KHOÁ NGOẠI VÀ RÀNG BUỘC CHECK KIỂM TRA HỢP LỆ CỦA MẬT KHẨU (ĐỘ DÀI 8 KÍ TỰ TRỞ LÊN)**

ALTER TABLE DANG\_NHAP

ADD CONSTRAINT fk\_manhanvien\_dangnhap\_nhanvien FOREIGN KEY (MaNhanVien) REFERENCES NHAN\_VIEN(MaNhanVien),

CONSTRAINT check\_matkhau\_dangnhap CHECK (len(MatKhau)>7);

**-- 4.2.9 CÁC RÀNG BUỘC KHOÁ NGOẠI**

ALTER TABLE PHIEU\_MUON

ADD CONSTRAINT fk\_madocgia\_phieumuon\_docgia FOREIGN KEY (MaDocGia) REFERENCES DOC\_GIA(MaDocGia) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_manhanvien\_phieumuon\_nhanvien FOREIGN KEY (MaNhanVien) REFERENCES NHAN\_VIEN(MaNhanVien) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT default\_maphieumuon DEFAULT dbo.Auto\_MaPhieuMuon() FOR MaPhieuMuon;

**-- 4.2.10 CÁC RÀNG BUỘC KHOÁ NGOẠI**

ALTER TABLE CHI\_TIET\_MUON\_TRA

ADD CONSTRAINT fk\_maphieumuon\_chitietphieumuon\_phieumuon FOREIGN KEY (MaPhieuMuon) REFERENCES PHIEU\_MUON(MaPhieuMuon) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_masach\_chitietphieumuon\_sach FOREIGN KEY (MaSach) REFERENCES SACH(MaSach) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_manhanvien\_chitietmuontra\_nhanvien FOREIGN KEY (MaNhanVienTra) REFERENCES NHAN\_VIEN(MaNhanVien),

CONSTRAINT fk\_matinhtrangsach\_chitietmuontra\_tinhtrangsach FOREIGN KEY (MaTinhTrangSach) REFERENCES TINH\_TRANG\_SACH(MaTinhTrangSach) ON DELETE CASCADE ON UPDATE CASCADE;

## 4.3 Trigger

**-- 4.3.1 Trigger tăng giá trị SoLuong trong PHIEUMUON sau khi Insert giá trị có MaPhieuMuon tương ứng vào CHITIETPHIEUMUON**

CREATE TRIGGER [TRIGGER\_THEM\_SL\_SACH\_PHIEU\_MUON]

ON CHI\_TIET\_MUON\_TRA AFTER INSERT

AS

BEGIN

UPDATE PHIEU\_MUON

SET SoLuong = SoLuong + 1

FROM inserted

WHERE [PHIEU\_MUON].MaPhieuMuon = inserted.MaPhieuMuon

END

GO

**-- 4.3.2 Trigger giảm giá trị SoLuong trong PHIEUMUON sau khi Delete giá trị có MaPhieuMuon tương ứng trong CHITIETPHIEUMUON**

CREATE TRIGGER [TRIGGER\_GIAM\_SL\_SACH\_PHIEU\_MUON]

ON CHI\_TIET\_MUON\_TRA AFTER DELETE

AS

BEGIN

UPDATE PHIEU\_MUON

SET SoLuong = SoLuong - 1

FROM deleted

WHERE [PHIEU\_MUON].MaPhieuMuon = deleted.MaPhieuMuon

END

GO

**-- 4.3.3 Trigger giảm SoLuong trong Sach sau khi sách đó bị mượn**

CREATE TRIGGER [TRIGGER\_SL\_SACH\_SAU\_MUON]

ON [CHI\_TIET\_MUON\_TRA] AFTER INSERT

AS

BEGIN

UPDATE SACH

SET SoLuong = SoLuong - 1

FROM inserted

WHERE [SACH].MaSach = inserted.MaSach

END

GO

**-- 4.3.4 Trigger tăng SoLuong trong Sach sau khi sách đó được trả lại**

CREATE TRIGGER [TRIGGER\_SL\_SACH\_SAU\_TRA]

ON [CHI\_TIET\_MUON\_TRA] AFTER UPDATE

AS IF (UPDATE(NgayTra))

BEGIN

UPDATE SACH

SET SoLuong = SoLuong + 1

FROM inserted

WHERE [SACH].MaSach = inserted.MaSach

END

GO

**-- 4.3.5 Trigger kiểm tra điều kiện khi mượn xem độc giả có đang vi phạm quy định mượn trả nào không**

CREATE TRIGGER [TRIGGER\_KIEM\_TRA\_DIEU\_KIEN\_MUON]

ON PHIEU\_MUON FOR UPDATE

AS IF (UPDATE(SoLuong))

BEGIN

DECLARE @MaPM varchar(10), @MaLoaiDT int, @GioiHanGT int, @GioiHanSTK int

SELECT @MaPM = MaPhieuMuon FROM inserted

-- Kiểm tra nếu có sách trễ hạn chưa trả

IF ((SELECT COUNT(\*) FROM [CHI\_TIET\_MUON\_TRA] ctmt

JOIN inserted i ON i.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [DOC\_GIA] dg ON i.MaDocGia = dg.MaDocGia

WHERE dg.MaDocGia = i.MaDocGia AND NgayTra IS NULL

AND GETDATE()>ctmt.NgayHetHan)>0)

BEGIN

DELETE FROM PHIEU\_MUON WHERE MaPhieuMuon = @MaPM

PRINT 'Có sách trễ hạn chưa trả'

END

-- Gán giới hạn sách tham khảo và giáo trình đối với từng đối tượng

SELECT @MaLoaiDT = dg.MaDoiTuong FROM inserted i JOIN [DOC\_GIA] dg ON i.MaDocGia = dg.MaDocGia

IF (@MaLoaiDT = 1 OR @MaLoaiDT = 3)

BEGIN

SELECT @GioiHanSTK = 10

SELECT @GioiHanGT = 15

END

ELSE IF (@MaLoaiDT = 2)

BEGIN

SELECT @GioiHanSTK = 10

SELECT @GioiHanGT = 20

END

ELSE IF (@MaLoaiDT = 4)

BEGIN

SELECT @GioiHanSTK = 5

SELECT @GioiHanGT = 5

END

ELSE IF (@MaLoaiDT = 5)

BEGIN

SELECT @GioiHanSTK = 10

SELECT @GioiHanGT = 5

END

-- Kiểm tra nếu số lượng sách mượn vượt quá quy định

IF ((SELECT COUNT(\*) FROM [CHI\_TIET\_MUON\_TRA] ctmt

JOIN inserted i ON i.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [DOC\_GIA] dg ON i.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

WHERE dg.MaDocGia = i.MaDocGia AND NgayTra IS NULL

AND s.LoaiSach = 1)>@GioiHanSTK)

BEGIN

DELETE FROM PHIEU\_MUON WHERE MaPhieuMuon = @MaPM

PRINT N'Vượt quá sách tham khảo'

END

IF ((SELECT COUNT(\*) FROM [CHI\_TIET\_MUON\_TRA] ctmt

JOIN inserted i ON i.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [DOC\_GIA] dg ON i.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

WHERE dg.MaDocGia = i.MaDocGia AND NgayTra IS NULL

AND s.LoaiSach = 0)>@GioiHanGT)

BEGIN

DELETE FROM PHIEU\_MUON WHERE MaPhieuMuon = @MaPM

PRINT N'Vượt quá giáo trình'

END

-- Kiểm tra nếu có sách đang mượn trùng lặp

IF ((SELECT COUNT(\*) FROM [CHI\_TIET\_MUON\_TRA] ctmt

JOIN inserted i ON i.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [DOC\_GIA] dg ON i.MaDocGia = dg.MaDocGia

WHERE dg.MaDocGia = i.MaDocGia AND NgayTra IS NULL

GROUP BY ctmt.MaSach HAVING COUNT(\*)>1)>1)

BEGIN

DELETE FROM PHIEU\_MUON WHERE MaPhieuMuon = @MaPM

PRINT N'2 sách trùng lặp'

END

END

GO

**-- Trigger tạo một user trên SQL khi thêm một nhân viên**

CREATE TRIGGER TRIGGER\_SQL\_ACCOUNT ON DANG\_NHAP

AFTER INSERT

AS

DECLARE @userName nvarchar(20), @passWord nvarchar(20)

SELECT @userName=i.TenDangNhap, @passWord=i.MatKhau

FROM inserted i

BEGIN

DECLARE @sqlString nvarchar(2000)

SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+ @passWord +''',

DEFAULT\_DATABASE=[QUANLYTHUVIEN], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName

EXEC (@sqlString)

END

GO

**-- 4.3.6 Trigger tạo một user trên SQL khi thêm một nhân viên**

CREATE TRIGGER TRIGGER\_SQL\_ACCOUNT ON DANG\_NHAP

AFTER INSERT

AS

DECLARE @userName nvarchar(20), @passWord nvarchar(20)

SELECT @userName=i.TenDangNhap, @passWord=i.MatKhau

FROM inserted i

BEGIN

DECLARE @sqlString nvarchar(2000)

SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+ @passWord +''',

DEFAULT\_DATABASE=[QUANLYTHUVIEN], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName

EXEC (@sqlString)

END

GO

**-- 4.3.7 Trigger khi xóa nhân viên thì sẽ đánh dấu nhân viên nghỉ và xóa tài khoản đăng nhập của người đó khỏi hệ thống**

CREATE TRIGGER XoaNhanVien ON NHAN\_VIEN

INSTEAD OF DELETE

AS

DECLARE @MaNV varchar(10)

SELECT @MaNV= d.MaNhanVien

FROM deleted d

SET XACT\_ABORT ON

BEGIN TRAN

BEGIN TRY

-- Cập nhật tình trạng tình việc của nhân viên là đã nghỉ

UPDATE NHAN\_VIEN SET TinhTrangLamViec=0 WHERE MaNhanVien=@maNV

DECLARE @TenDangNhap varchar(20)

SELECT @TenDangNhap=TenDangNhap FROM DANG\_NHAP WHERE MaNhanVien=@MaNV

-- Xóa user của nhân viên

DECLARE @sql varchar(100)

SET @sql = 'DROP USER '+ @TenDangNhap

EXEC @sql

-- Xóa tài khoản login của nhân viên khỏi server

SET @sql = 'DROP LOGIN '+ @TenDangNhap

EXEC @sql

-- Xóa tài khoản đăng nhập của nhân viên khỏi bảng Đăng nhập

DELETE FROM DANG\_NHAP WHERE MaNhanVien=@MaNV;

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

END CATCH

GO

## 4.4 View

**-- 4.4.1 VIEW CHI TIẾT THÔNG TIN SÁCH**

CREATE VIEW [VIEW\_CHI\_TIET\_SACH] AS

SELECT s.MaSach [Mã sách], s.TenSach [Tên sách],

CASE WHEN LoaiSach = 1 THEN 'Sách tham khảo' ELSE N'Giáo trình' END [Loại sách],

(SELECT STRING\_AGG([TenTacGia], ' - ') FROM [TAC\_GIA] tg, [SACH\_TAC\_GIA] stg

WHERE tg.MaTacGia = stg.MaTacGia AND s.MaSach = stg.MaSach) [Tác giả],

nxb.TenNhaXuatBan [Nhà xuất bản], cn.TenChuyenNganh [Chuyên ngành], s.GiaBia [Giá bìa], s.SoLuong [Số lượng]

FROM [SACH] s JOIN [NHA\_XUAT\_BAN] nxb ON s.MaNhaXuatBan = nxb.MaNhaXuatBan

JOIN [CHUYEN\_NGANH] cn ON s.MaChuyenNganh = cn.MaChuyenNganh

**-- 4.4.2 VIEW CHI TIẾT THÔNG TIN NHÂN VIÊN**

CREATE VIEW [VIEW\_CHI\_TIET\_NHAN\_VIEN] AS

SELECT nv.MaNhanVien [Mã nhân viên], HoTen [Họ tên],

CASE WHEN GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính],

NgaySinh [Ngày sinh], SoDienThoai [SĐT], Email, NgayVaoLam [Ngày vào làm],

CASE WHEN TinhTrangLamViec = 1 THEN N'Còn làm việc' ELSE N'Đã nghỉ việc' END [Tình trạng làm việc],

CASE WHEN dn.LoaiTaiKhoan = 1 THEN N'Thủ thư' ELSE N'Quản trị viên' END [Vai trò hệ thống]

FROM [NHAN\_VIEN] nv JOIN [DANG\_NHAP] dn ON nv.MaNhanVien = dn.MaNhanVien

**-- 4.4.3 VIEW CHI TIẾT THÔNG TIN ĐỘC GIẢ**

CREATE VIEW [VIEW\_CHI\_TIET\_THE\_DOC\_GIA] AS

SELECT MaDocGia [Mã độc giả], HoTen [Họ tên], TenDoiTuong [Đối tượng],

CASE WHEN dg.GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính], NgaySinh [Ngày sinh], SoDienThoai [SĐT], Email, NgayLamThe [Ngày làm thẻ], NgayHetHan [Ngày hết hạn]

FROM [DOC\_GIA] dg JOIN [DOI\_TUONG\_DOC\_GIA] ddg ON dg.MaDoiTuong = ddg.MaDoiTuong

**-- 4.4.4 VIEW CHI TIẾT THÔNG TIN CÁC SÁCH MƯỢN TRẢ**

CREATE VIEW [VIEW\_CHI\_TIET\_MUON\_TRA] AS

SELECT pm.MaPhieuMuon [Mã phiếu mượn], pm.MaDocGia [Mã người mượn],

dg.HoTen [Họ tên], s.MaSach [Mã sách], TenSach [Tên sách], NgayMuon [Ngày mượn],

ctmt.NgayHetHan [Ngày hết hạn], NgayTra AS [Ngày trả], nv.HoTen [Nhân viên cho mượn],

tts.TinhTrangSach [Tình trạng sách], PhatHuHong [Phạt hư hỏng], PhatQuaHan [Phạt quá hạn]

FROM [CHI\_TIET\_MUON\_TRA] ctmt JOIN [PHIEU\_MUON] pm ON ctmt.MaPhieuMuon = pm.MaPhieuMuon

JOIN [DOC\_GIA] dg ON pm.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

JOIN [NHAN\_VIEN] nv ON nv.MaNhanVien = pm.MaNhanVien

LEFT OUTER JOIN [TINH\_TRANG\_SACH] tts ON ctmt.MaTinhTrangSach = tts.MaTinhTrangSach

**-- 4.4.5 VIEW THÔNG TIN CÁC SÁCH ĐANG ĐƯỢC MƯỢN**

CREATE VIEW [VIEW\_SACH\_DANG\_MUON] AS

SELECT pm.MaPhieuMuon [Mã phiếu mượn], pm.MaDocGia [Mã người mượn],

dg.HoTen [Họ tên], s.MaSach [Mã sách], s.TenSach [Tên sách],

pm.NgayMuon [Ngày mượn], ctmt.NgayHetHan [Ngày hết hạn], nv.HoTen [Nhân viên cho mượn]

FROM [CHI\_TIET\_MUON\_TRA] ctmt JOIN [PHIEU\_MUON] pm ON ctmt.MaPhieuMuon = pm.MaPhieuMuon

JOIN [DOC\_GIA] dg ON pm.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

JOIN [NHAN\_VIEN] nv ON nv.MaNhanVien = pm.MaNhanVien

WHERE NgayTra IS NULL

**-- 4.4.6 VIEW THÔNG TIN CÁC SÁCH TRỄ HẠN CHƯA TRẢ**

CREATE VIEW [VIEW\_SACH\_TRE\_HAN] AS

SELECT pm.MaPhieuMuon [Mã phiếu mượn], pm.MaDocGia [Mã người mượn],

dg.HoTen [Họ tên], s.MaSach [Mã sách], s.TenSach [Tên sách],

pm.NgayMuon [Ngày mượn], ctmt.NgayHetHan [Ngày hết hạn], nv.HoTen [Nhân viên cho mượn]

FROM [CHI\_TIET\_MUON\_TRA] ctmt JOIN [PHIEU\_MUON] pm ON ctmt.MaPhieuMuon = pm.MaPhieuMuon

JOIN [DOC\_GIA] dg ON pm.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

JOIN [NHAN\_VIEN] nv ON nv.MaNhanVien = pm.MaNhanVien

WHERE (GETDATE()>ctmt.NgayHetHan) AND NgayTra IS NULL

**-- 4.4.7 VIEW THÔNG TIN CÁC SÁCH ĐÃ TRẢ**

CREATE VIEW [VIEW\_SACH\_DA\_TRA] AS

SELECT pm.MaPhieuMuon [Mã phiếu mượn], pm.MaDocGia [Mã người mượn],

dg.HoTen [Họ tên], s.MaSach [Mã sách], TenSach [Tên sách], NgayMuon [Ngày mượn],

ctmt.NgayHetHan [Ngày hết hạn], NgayTra AS [Ngày trả], nv.HoTen [Nhân viên cho mượn],

tts.TinhTrangSach [Tình trạng sách], PhatHuHong [Phạt hư hỏng], PhatQuaHan [Phạt quá hạn]

FROM [CHI\_TIET\_MUON\_TRA] ctmt JOIN [PHIEU\_MUON] pm ON ctmt.MaPhieuMuon = pm.MaPhieuMuon

JOIN [DOC\_GIA] dg ON pm.MaDocGia = dg.MaDocGia

JOIN [SACH] s ON s.MaSach = ctmt.MaSach

JOIN [NHAN\_VIEN] nv ON nv.MaNhanVien = pm.MaNhanVien

JOIN [TINH\_TRANG\_SACH] tts ON ctmt.MaTinhTrangSach = tts.MaTinhTrangSach

WHERE NgayTra IS NOT NULL

**-- 4.4.8 VIEW ĐỘC GIẢ CÒN HẠN THẺ**

CREATE VIEW [VIEW\_DOC\_GIA\_CON\_HAN] AS

SELECT MaDocGia [Mã độc giả], HoTen [Họ tên], TenDoiTuong [Đối tượng],

CASE WHEN dg.GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính], NgaySinh [Ngày sinh],

SoDienThoai [SĐT], Email, NgayLamThe [Ngày làm thẻ], NgayHetHan [Ngày hết hạn]

FROM [DOC\_GIA] dg JOIN [DOI\_TUONG\_DOC\_GIA] ddg ON dg.MaDoiTuong = ddg.MaDoiTuong

WHERE GETDATE()<NgayHetHan

**-- 4.4.9 VIEW NHÂN VIÊN CÒN LÀM VIỆC**

CREATE VIEW [VIEW\_NHAN\_VIEN\_LAM\_VIEC] AS

SELECT nv.MaNhanVien [Mã nhân viên], HoTen [Họ tên],

CASE WHEN GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính],

NgaySinh [Ngày sinh], SoDienThoai [SĐT], Email, NgayVaoLam [Ngày vào làm],

CASE WHEN dn.LoaiTaiKhoan = 1 THEN N'Thủ thư' ELSE N'Quản trị viên' END [Vai trò hệ thống]

FROM [NHAN\_VIEN] nv JOIN [DANG\_NHAP] dn ON nv.MaNhanVien = dn.MaNhanVien

WHERE TinhTrangLamViec = 1

GO

**-- 4.4.10 VIEW THÔNG TIN CÁC ĐỘC GIẢ ĐANG MƯỢN SÁCH**

CREATE VIEW [VIEW\_DOC\_GIA\_DANG\_MUON\_SACH] AS

SELECT DISTINCT dg.MaDocGia [Độc giả], HoTen [Họ tên], NgaySinh [Ngày sinh],

CASE WHEN GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính], SoDienThoai [SĐT], Email

FROM [DOC\_GIA] dg JOIN [PHIEU\_MUON] pm ON dg.MaDocGia = pm.MaDocGia

JOIN [CHI\_TIET\_MUON\_TRA] ctmt ON pm.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [SACH] s ON ctmt.MaSach = s.MaSach

WHERE NgayTra IS NULL

GO

**-- 4.4.11 VIEW CÁC ĐỘC GIẢ ĐÃ TRỄ HẠN TRẢ SÁCH**

CREATE VIEW [VIEW\_DOC\_GIA\_DANG\_TRE\_HAN] AS

SELECT DISTINCT dg.MaDocGia [Độc giả], HoTen [Họ tên], NgaySinh [Ngày sinh],

CASE WHEN GioiTinh = 1 THEN 'Nam' ELSE N'Nữ' END [Giới tính], SoDienThoai [SĐT], Email

FROM [DOC\_GIA] dg JOIN [PHIEU\_MUON] pm ON dg.MaDocGia = pm.MaDocGia

JOIN [CHI\_TIET\_MUON\_TRA] ctmt ON pm.MaPhieuMuon = ctmt.MaPhieuMuon

JOIN [SACH] s ON ctmt.MaSach = s.MaSach

WHERE NgayTra IS NULL AND GETDATE()>ctmt.NgayHetHan

GO

# CHƯƠNG 3. THIẾT KẾ CÁC CHỨC NĂNG

## 0. Kết nối cơ sở dữ liệu

- Hàm kết nối nằm trong *DbHelper.cs*:

public static SqlConnection Connect()

{

String connectionString = @"Data Source =(local);Initial Catalog=QuanLyThuVien; User ID="+Username+";Password="+Password;

SqlConnection conn = new SqlConnection(connectionString);

return conn;

}

- Tài khoản đăng nhập vào hệ thống:

|  |  |  |
| --- | --- | --- |
| **Vai trò** | **Tên đăng nhập** | **Mật khẩu** |
| admin | admin | admin123 |
| nhanvien | hoangtrung | trung123 |

## Procedure

### 1.1 Quản lý Sách

**-- 1.1.1 THÊM SÁCH (VÀ THÊM TÁC GIẢ CỦA SÁCH VÀ BẢNG SACH\_TAC\_GIA)**

CREATE OR ALTER PROCEDURE THEM\_SACH (@TenSach nvarchar(100), @LoaiSach bit, @MaNhaXuatBan varchar(10), @MaChuyenNganh varchar(10), @GiaBia decimal(9,3), @SoLuong int, @MaTacGia1 varchar(10), @MaTacGia2 varchar(10), @MaTacGia3 varchar(10))

AS

BEGIN

SET XACT\_ABORT ON

BEGIN TRY

BEGIN TRAN

IF (@MaTacGia1 IS NULL AND @MaTacGia2 IS NULL AND @MaTacGia3 IS NULL)

BEGIN

RAISERROR(N'Chưa có tác giả!!', 16,1)

RETURN

END

IF (@GiaBia < 0)

BEGIN

RAISERROR(N'Giá bìa không hợp lệ!!', 16,1)

RETURN

END

IF (@SoLuong < 0)

BEGIN

RAISERROR(N'Số lượng không hợp lệ!!', 16,1)

RETURN

END

DECLARE @MaSach varchar(10) = dbo.Auto\_MaSach()

INSERT INTO SACH (TenSach, LoaiSach, MaNhaXuatBan, MaChuyenNganh, GiaBia, SoLuong)

VALUES (@TenSach, @LoaiSach, @MaNhaXuatBan, @MaChuyenNganh, @GiaBia, @SoLuong)

IF (@MaTacGia1 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia1)

IF (@MaTacGia2 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia2)

IF (@MaTacGia3 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia3)

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE THEM\_SACH VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("THEM\_SACH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value = tbTenSach.Text.ToString();

if (rbGiaoTrinh.Checked)

cmd.Parameters.Add("@LoaiSach", SqlDbType.Bit).Value = 0;

else

cmd.Parameters.Add("@LoaiSach", SqlDbType.Bit).Value = 1;

cmd.Parameters.Add("@MaNhaXuatBan", SqlDbType.VarChar).Value = cbNhaXuatBan.SelectedValue;

cmd.Parameters.Add("@MaChuyenNganh", SqlDbType.VarChar).Value = cbChuyenNganh.SelectedValue;

cmd.Parameters.Add("@GiaBia", SqlDbType.Decimal).Value = tbGiaBia.Text.ToString();

cmd.Parameters.Add("@SoLuong", SqlDbType.Int).Value = tbSoLuong.Text.ToString();

if (cbTacGia1.SelectedIndex == -1)

cmd.Parameters.Add("@MaTacGia1", SqlDbType.VarChar).Value = DBNull.Value;

else

cmd.Parameters.Add("@MaTacGia1", SqlDbType.VarChar).Value = cbTacGia1.SelectedValue;

if (cbTacGia2.SelectedIndex == -1)

cmd.Parameters.Add("@MaTacGia2", SqlDbType.VarChar).Value = DBNull.Value;

else

cmd.Parameters.Add("@MaTacGia2", SqlDbType.VarChar).Value = cbTacGia2.SelectedValue;

if (cbTacGia3.SelectedIndex == -1)

cmd.Parameters.Add("@MaTacGia3", SqlDbType.VarChar).Value = DBNull.Value;

else

cmd.Parameters.Add("@MaTacGia3", SqlDbType.VarChar).Value = cbTacGia3.SelectedValue;

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.1.2 SỬA SÁCH (VÀ SỬA TÁC GIẢ CỦA SÁCH TRONG BẢNG SACH\_TAC\_GIA)**

CREATE OR ALTER PROCEDURE SUA\_SACH (@MaSach varchar(10), @TenSach nvarchar(100), @LoaiSach bit, @MaNhaXuatBan varchar(10), @MaChuyenNganh varchar(10), @GiaBia decimal(9,3), @SoLuong int, @MaTacGia1 varchar(10), @MaTacGia2 varchar(10), @MaTacGia3 varchar(10))

AS

BEGIN

BEGIN TRY

BEGIN TRAN

IF (@MaTacGia1 IS NULL AND @MaTacGia2 IS NULL AND @MaTacGia3 IS NULL)

BEGIN

RAISERROR(N'Chưa có tác giả!!', 16,1)

RETURN

END

IF (@GiaBia < 0)

BEGIN

RAISERROR(N'Giá bìa không hợp lệ!!', 16,1)

RETURN

END

IF (@SoLuong < 0)

BEGIN

RAISERROR(N'Số lượng không hợp lệ!!', 16,1)

RETURN

END

DELETE SACH\_TAC\_GIA WHERE MaSach = @MaSach

IF (@MaTacGia1 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia1)

IF (@MaTacGia2 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia2)

IF (@MaTacGia3 IS NOT NULL)

INSERT INTO SACH\_TAC\_GIA (MaSach, MaTacGia) VALUES (@MaSach, @MaTacGia3)

UPDATE SACH

SET TenSach = @TenSach, LoaiSach = @LoaiSach, MaNhaXuatBan = @MaNhaXuatBan, MaChuyenNganh = @MaChuyenNganh, GiaBia = @GiaBia, SoLuong = @SoLuong

WHERE MaSach = @MaSach

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE SUA\_SACH VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("SUA\_SACH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = tbMaSach.Text.ToString();

cmd.Parameters.Add("@TenSach", SqlDbType.NVarChar).Value = tbTenSach.Text.ToString();

if (rbGiaoTrinh.Checked == true)

cmd.Parameters.Add("@LoaiSach", SqlDbType.Bit).Value = 0;

else

cmd.Parameters.Add("@LoaiSach", SqlDbType.Bit).Value = 1;

cmd.Parameters.Add("@MaNhaXuatBan", SqlDbType.VarChar).Value = cbNhaXuatBan.SelectedValue;

cmd.Parameters.Add("@MaChuyenNganh", SqlDbType.VarChar).Value = cbChuyenNganh.SelectedValue;

cmd.Parameters.Add("@GiaBia", SqlDbType.Decimal).Value = tbGiaBia.Text.ToString();

cmd.Parameters.Add("@SoLuong", SqlDbType.Int).Value = tbSoLuong.Text.ToString();

cmd.Parameters.Add("@MaTacGia1", SqlDbType.VarChar).Value = cbTacGia1.SelectedValue;

if (cbTacGia2.SelectedIndex == -1)

cmd.Parameters.Add("@MaTacGia2", SqlDbType.VarChar).Value = DBNull.Value;

else

cmd.Parameters.Add("@MaTacGia2", SqlDbType.VarChar).Value = cbTacGia2.SelectedValue;

if (cbTacGia3.SelectedIndex == -1)

cmd.Parameters.Add("@MaTacGia3", SqlDbType.VarChar).Value = DBNull.Value;

else

cmd.Parameters.Add("@MaTacGia3", SqlDbType.VarChar).Value = cbTacGia3.SelectedValue;

cmd.ExecuteNonQuery();

} catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.1.3 XOÁ SÁCH**

CREATE OR ALTER PROCEDURE XOA\_SACH (@MaSach varchar(10))

AS

BEGIN

DELETE SACH WHERE MaSach = @MaSach

END

GO

**// THỰC THI PROCEDURE XOA\_SACH**

SqlCommand cmd = new SqlCommand("XOA\_SACH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = tbMaSach.Text.ToString();

cmd.ExecuteNonQuery();

### 1.2 Quản lý độc giả

**-- 1.2.1 THÊM ĐỘC GIẢ**

CREATE OR ALTER PROCEDURE THEM\_DOC\_GIA (@MaDocGia varchar(10), @MaDoiTuong int, @HoTen nvarchar(50), @GioiTinh bit, @NgaySinh date, @SDT varchar(10), @Email varchar(50))

AS

BEGIN

BEGIN TRY

IF (DATEDIFF(year, @NgaySinh, GETDATE())<18)

BEGIN

RAISERROR(N'Độc giả chưa đủ 18 tuổi!!', 16,1)

RETURN

END

IF (len(@SDT)<>10)

BEGIN

RAISERROR(N'Số điện thoại không hợp lệ!!', 16,1)

RETURN

END

IF (@Email NOT LIKE '%\_\_\_@\_\_\_%.\_\_%')

BEGIN

RAISERROR(N'Email không hợp lệ!!', 16,1)

RETURN

END

INSERT INTO DOC\_GIA (MaDocGia, MaDoiTuong, HoTen, GioiTinh, NgaySinh, SoDienThoai, Email, NgayLamThe, NgayHetHan)

VALUES (@MaDocGia, @MaDoiTuong, @HoTen, @GioiTinh, @NgaySinh, @SDT, @Email, GETDATE(), DATEADD(yyyy, 1, GETDATE()))

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE THEM\_DOC\_GIA VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("THEM\_DOC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaDocGia", SqlDbType.VarChar).Value = tbMaDocGia.Text.ToString();

if (rbNam.Checked)

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 1;

else

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 0;

cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = tbHoTen.Text.ToString();

cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = tbEmail.Text.ToString();

cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = tbSDT.Text.ToString();

cmd.Parameters.Add("@MaDoiTuong", SqlDbType.Int).Value = cbDoiTuong.SelectedValue;

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = dateTimePicker.Value.Date;

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.2.2 SỬA ĐỘC GIẢ**

CREATE OR ALTER PROCEDURE SUA\_DOC\_GIA (@MaDocGia varchar(10), @MaDoiTuong int, @HoTen nvarchar(50), @GioiTinh bit, @NgaySinh date,

@SDT varchar(10), @Email varchar(50))

AS

BEGIN

BEGIN TRY

IF (DATEDIFF(year, @NgaySinh, GETDATE())<18)

BEGIN

RAISERROR(N'Độc giả chưa đủ 18 tuổi!!', 16,1)

RETURN

END

IF (len(@SDT)<>10)

BEGIN

RAISERROR(N'Số điện thoại không hợp lệ!!', 16,1)

RETURN

END

IF (@Email NOT LIKE '%\_\_\_@\_\_\_%.\_\_%')

BEGIN

RAISERROR(N'Email không hợp lệ!!', 16,1)

RETURN

END

UPDATE DOC\_GIA

SET MaDoiTuong = @MaDoiTuong, HoTen = @HoTen, GioiTinh = @GioiTinh, NgaySinh = @NgaySinh, SoDienThoai = @SDT, Email = @Email

WHERE MaDocGia = @MaDocGia

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE SUA\_DOC\_GIA VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("SUA\_DOC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaDocGia", SqlDbType.VarChar).Value = tbMaDocGia.Text.ToString();

cmd.Parameters.Add("@MaDoiTuong", SqlDbType.Int).Value = cbDoiTuong.SelectedValue;

cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = tbTenDocGia.Text.ToString();

if (cbGioiTinh.SelectedItem.ToString() == "Nữ")

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 0;

else

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 1;

DateTime date = DateTime.ParseExact(tbNgaySinh.Text, "dd/MM/yyyy", null);

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = date;

cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = tbSDT.Text.ToString();

cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = tbEmail.Text.ToString();

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.2.3 XOÁ ĐỘC GIẢ**

CREATE OR ALTER PROCEDURE XOA\_DOC\_GIA (@MaDocGia varchar(10))

AS

BEGIN

DELETE DOC\_GIA WHERE MaDocGia = @MaDocGia

END

GO

**// THỰC THI PROCEDURE XOA\_DOC\_GIA**

SqlCommand cmd = new SqlCommand("XOA\_DOC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaDocGia", SqlDbType.VarChar).Value = tbMaDocGia.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.2.4 GIA HẠN THẺ ĐỘC GIẢ (1 NĂM)**

CREATE OR ALTER PROCEDURE GIA\_HAN (@MaDocGia varchar(10))

AS

BEGIN

UPDATE DOC\_GIA

SET NgayHetHan = DATEADD(yyyy, 1, NgayHetHan)

WHERE MaDocGia = @MaDocGia

END

GO

**// THỰC THI PROCEDURE GIA\_HAN**

SqlCommand cmd = new SqlCommand("GIA\_HAN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaDocGia", SqlDbType.VarChar).Value = tbMaDocGia.Text.ToString();

cmd.ExecuteNonQuery();

### 1.3 Quản lý Mượn trả

**-- 1.3.1 THÊM PHIẾU MƯỢN**

CREATE OR ALTER PROCEDURE THEM\_PHIEU\_MUON (@MaDocGia varchar(10), @MaNhanVien varchar(10))

AS

BEGIN

INSERT INTO PHIEU\_MUON(MaDocGia, MaNhanVien, NgayMuon, SoLuong)

VALUES (@MaDocGia, @MaNhanVien, GETDATE(), 0)

END

GO

**-- 1.3.2 THÊM SÁCH MƯỢN**

CREATE OR ALTER PROCEDURE THEM\_SACH\_MUON (@MaPhieuMuon varchar(10), @MaSach varchar(10))

AS

BEGIN

BEGIN TRY

IF ((SELECT SoLuong FROM SACH WHERE MaSach = @MaSach) = 0)

BEGIN

RAISERROR('Không thể mượn. Sách được chọn đã hết!!', 16, 1)

RETURN

END

-- (Các điều kiện khác về vi phạm quy định mượn được kiểm tra và raiserror trong trigger)

DECLARE @DoiTuong int = (SELECT dg.MaDoiTuong

FROM PHIEU\_MUON pm JOIN DOC\_GIA dg ON pm.MaDocGia = dg.MaDocGia

WHERE pm.MaPhieuMuon = @MaPhieuMuon)

DECLARE @LoaiSach bit = (SELECT LoaiSach FROM SACH WHERE MaSach = @MaSach)

DECLARE @NgayHetHan date

-- Tuỳ theo loại sách và đối tượng mượn sách,

-- thời gian hết hạn tự động được gia hạn theo quy định

IF (@LoaiSach = 1)

BEGIN

IF (@DoiTuong = 5)

SET @NgayHetHan = DATEADD(yyyy, 1, GETDATE())

ELSE

SET @NgayHetHan = DATEADD(dd, 28, GETDATE())

END

ELSE BEGIN

IF (@DoiTuong = 4)

SET @NgayHetHan = DATEADD(dd, 56, GETDATE())

ELSE IF (@DoiTuong = 5)

SET @NgayHetHan = DATEADD(yyyy, 1, GETDATE())

ELSE

SET @NgayHetHan = DATEADD(mm, 4, GETDATE())

END

INSERT INTO CHI\_TIET\_MUON\_TRA(MaPhieuMuon, MaSach, NgayHetHan)

VALUES (@MaPhieuMuon, @MaSach, @NgayHetHan)

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE THEM\_PHIEU\_MUON VÀ THEM\_SACH\_MUON VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("Auto\_MaPhieuMuon", conn);

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT [dbo].[Auto\_MaPhieuMuon]()";

var result = cmd.ExecuteScalar();

String maPhieuMuon = (String)result;

SqlCommand cmd1 = new SqlCommand("THEM\_PHIEU\_MUON", conn);

cmd1.CommandType = CommandType.StoredProcedure;

cmd1.Parameters.Add("@MaDocGia", SqlDbType.VarChar).Value = tbMaDocGia.Text;

cmd1.Parameters.Add("@MaNhanVien", SqlDbType.VarChar).Value = DbHelper.MaNhanVien;

cmd1.ExecuteNonQuery();

var labels = new List<Label>() { label1, label2, label3, label4, label5, label6, label7, label8, label9, label10 };

foreach (var label in labels)

{

if (label.Text != "")

{

SqlCommand cmd2 = new SqlCommand("THEM\_SACH\_MUON", conn);

cmd2.CommandType = CommandType.StoredProcedure;

cmd2.Parameters.Add("@MaPhieuMuon", SqlDbType.VarChar).Value = maPhieuMuon;

cmd2.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = label.Text;

cmd2.ExecuteNonQuery();

}

else break;

}

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.3.3 GIA HẠN THỜI GIAN MƯỢN SÁCH**

CREATE OR ALTER PROCEDURE GIA\_HAN\_SACH (@MaPhieuMuon varchar(10), @MaSach varchar(10))

AS

BEGIN

BEGIN TRY

IF (GETDATE()>(SELECT NgayHetHan FROM CHI\_TIET\_MUON\_TRA

WHERE MaPhieuMuon = @MaPhieuMuon AND MaSach = @MaSach))

BEGIN

RAISERROR('Sách đã trễ hạn, không thể gia hạn. Vui lòng trả sách!!', 16, 1)

RETURN

END

DECLARE @DoiTuong int = (SELECT dg.MaDoiTuong

FROM PHIEU\_MUON pm JOIN DOC\_GIA dg ON pm.MaDocGia = dg.MaDocGia

WHERE pm.MaPhieuMuon = @MaPhieuMuon)

DECLARE @LoaiSach bit = (SELECT LoaiSach FROM SACH WHERE MaSach = @MaSach)

DECLARE @NgayHetHan date

IF (@LoaiSach = 1)

BEGIN

IF (@DoiTuong = 5)

SET @NgayHetHan = DATEADD(yyyy, 1, GETDATE())

ELSE

SET @NgayHetHan = DATEADD(dd, 28, GETDATE())

END

ELSE BEGIN

IF (@DoiTuong = 4)

SET @NgayHetHan = DATEADD(dd, 56, GETDATE())

ELSE IF (@DoiTuong = 5)

SET @NgayHetHan = DATEADD(yyyy, 1, GETDATE())

ELSE

SET @NgayHetHan = DATEADD(mm, 4, GETDATE())

END

UPDATE CHI\_TIET\_MUON\_TRA

SET NgayHetHan = @NgayHetHan

WHERE MaPhieuMuon = @MaPhieuMuon AND MaSach = @MaSach

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE GIA\_HAN\_SACH VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("GIA\_HAN\_SACH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaPhieuMuon", SqlDbType.VarChar).Value = tbMaPhieuMuon.Text;

cmd.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = tbMaSach.Text;

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.3.4 TRẢ SÁCH**

CREATE OR ALTER PROCEDURE TRA\_SACH (@MaPhieuMuon varchar(10), @MaNhanVienTra varchar(10), @TinhTrang int, @MaSach varchar(10), @PhatHuHong decimal, @PhatQuaHan decimal)

AS

BEGIN

BEGIN TRY

IF ((SELECT NgayTra FROM CHI\_TIET\_MUON\_TRA

WHERE MaPhieuMuon = @MaPhieuMuon AND MaSach = @MaSach) IS NOT NULL)

BEGIN

RAISERROR('Sách này đã được trả rồi. Vui lòng kiểm tra lại!!', 16, 1)

RETURN

END

UPDATE CHI\_TIET\_MUON\_TRA

SET NgayTra = GETDATE(), MaTinhTrangSach = @TinhTrang, MaNhanVienTra = @MaNhanVienTra, PhatHuHong = @PhatHuHong, PhatQuaHan = @PhatQuaHan

WHERE MaPhieuMuon = @MaPhieuMuon AND MaSach = @MaSach

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("TRA\_SACH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaPhieuMuon", SqlDbType.VarChar).Value = FormMuonTraSach.maPhieuMuon;

cmd.Parameters.Add("@MaNhanVienTra", SqlDbType.VarChar).Value = DbHelper.MaNhanVien;

cmd.Parameters.Add("@TinhTrang", SqlDbType.Int).Value = cbTinhTrang.SelectedValue;

cmd.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = FormMuonTraSach.maSach;

cmd.Parameters.Add("@PhatHuHong", SqlDbType.Decimal).Value = phatHuHong;

cmd.Parameters.Add("@PhatQuaHan", SqlDbType.Decimal).Value = FormMuonTraSach.phatQuaHan;

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

### 1.4 Quản lý Tác giả

**-- 1.4.1 THÊM TÁC GIẢ**

CREATE OR ALTER PROCEDURE THEM\_TAC\_GIA (@TenTacGia nvarchar(50))

AS

BEGIN

INSERT INTO TAC\_GIA(TenTacGia) VALUES (@TenTacGia)

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("THEM\_TAC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenTacGia", SqlDbType.NVarChar).Value = tbTenThem.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.4.2 SỬA TÁC GIẢ**

CREATE OR ALTER PROCEDURE SUA\_TAC\_GIA (@MaTacGia varchar(10), @TenTacGia nvarchar(50))

AS

BEGIN

UPDATE TAC\_GIA

SET TenTacGia = @TenTacGia

WHERE MaTacGia = @MaTacGia

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("SUA\_TAC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaTacGia", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.Parameters.Add("@TenTacGia", SqlDbType.NVarChar).Value = tbTen.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.4.3 XOÁ TÁC GIẢ**

CREATE OR ALTER PROCEDURE XOA\_TAC\_GIA (@MaTacGia varchar(10))

AS

BEGIN

DELETE TAC\_GIA WHERE MaTacGia = @MaTacGia

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("XOA\_TAC\_GIA", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaTacGia", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.ExecuteNonQuery();

### 1.5 Quản lý Nhà xuất bản

**-- 1.5.1 THÊM NHÀ XUẤT BẢN**

CREATE OR ALTER PROCEDURE THEM\_NHA\_XUAT\_BAN (@TenNhaXuatBan nvarchar(50))

AS

BEGIN

INSERT INTO NHA\_XUAT\_BAN(TenNhaXuatBan) VALUES (@TenNhaXuatBan)

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("THEM\_NHA\_XUAT\_BAN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenNhaXuatBan", SqlDbType.NVarChar).Value = tbTenThem.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.5.2 SỬA NHÀ XUẤT BẢN**

CREATE OR ALTER PROCEDURE SUA\_NHA\_XUAT\_BAN (@MaNhaXuatBan varchar(10), @TenNhaXuatBan nvarchar(50))

AS

BEGIN

UPDATE NHA\_XUAT\_BAN

SET TenNhaXuatBan = @TenNhaXuatBan

WHERE MaNhaXuatBan = @MaNhaXuatBan

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("SUA\_NHA\_XUAT\_BAN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaNhaXuatBan", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.Parameters.Add("@TenNhaXuatBan", SqlDbType.NVarChar).Value = tbTen.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.5.3 XOÁ NHÀ XUẤT BẢN**

CREATE OR ALTER PROCEDURE XOA\_NHA\_XUAT\_BAN (@MaNhaXuatBan varchar(10))

AS

BEGIN

DELETE NHA\_XUAT\_BAN WHERE MaNhaXuatBan = @MaNhaXuatBan

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("XOA\_NHA\_XUAT\_BAN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaNhaXuatBan", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.ExecuteNonQuery();

### 1.6 Quản lý Chuyên ngành

**-- 1.6.1 THÊM CHUYÊN NGÀNH**

CREATE OR ALTER PROCEDURE THEM\_CHUYEN\_NGANH (@TenChuyenNganh nvarchar(50))

AS

BEGIN

INSERT INTO CHUYEN\_NGANH(TenChuyenNganh) VALUES (@TenChuyenNganh)

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("THEM\_CHUYEN\_NGANH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenChuyenNganh", SqlDbType.NVarChar).Value = tbTenThem.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.6.2 SỬA CHUYÊN NGÀNH**

CREATE OR ALTER PROCEDURE SUA\_CHUYEN\_NGANH (@MaChuyenNganh varchar(10), @TenChuyenNganh nvarchar(50))

AS

BEGIN

UPDATE CHUYEN\_NGANH

SET TenChuyenNganh = @TenChuyenNganh

WHERE MaChuyenNganh = @MaChuyenNganh

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("SUA\_CHUYEN\_NGANH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaChuyenNganh", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.Parameters.Add("@TenChuyenNganh", SqlDbType.NVarChar).Value = tbTen.Text.ToString();

cmd.ExecuteNonQuery();

**-- 1.6.3 XOÁ CHUYÊN NGÀNH**

CREATE OR ALTER PROCEDURE XOA\_CHUYEN\_NGANH (@MaChuyenNganh varchar(10))

AS

BEGIN

DELETE CHUYEN\_NGANH WHERE MaChuyenNganh = @MaChuyenNganh

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("XOA\_CHUYEN\_NGANH", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaChuyenNganh", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.ExecuteNonQuery();

### 1.7 Quản lý Nhân viên

**-- 1.7.1 THÊM NHÂN VIÊN (BAO GỒM CẢ THÊM TÀI KHOẢN ĐĂNG NHẬP)**

CREATE OR ALTER PROCEDURE THEM\_NHAN\_VIEN (@HoTen nvarchar(50), @GioiTinh bit, @NgaySinh date,

@SDT varchar(10), @Email varchar(50), @Username varchar(20), @Password varchar(20))

AS

BEGIN

BEGIN TRY

BEGIN TRAN

IF (DATEDIFF(year, @NgaySinh, GETDATE())<18)

BEGIN

RAISERROR(N'Độc giả chưa đủ 18 tuổi!!', 16,1)

RETURN

END

IF (len(@SDT)<>10)

BEGIN

RAISERROR(N'Số điện thoại không hợp lệ!!', 16,1)

RETURN

END

IF (@Email NOT LIKE '%\_\_\_@\_\_\_%.\_\_%')

BEGIN

RAISERROR(N'Email không hợp lệ!!', 16,1)

RETURN

END

DECLARE @MaNhanVien varchar(10) = dbo.Auto\_MaNhanVien()

INSERT INTO NHAN\_VIEN (HoTen, GioiTinh, NgaySinh, SoDienThoai, Email, NgayVaoLam)

VALUES (@HoTen, @GioiTinh, @NgaySinh, @SDT, @Email, GETDATE())

INSERT INTO DANG\_NHAP(TenDangNhap, MatKhau, MaNhanVien, LoaiTaiKhoan)

VALUES (@Username, @Password, @MaNhanVien, 1)

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("THEM\_NHAN\_VIEN", conn);

cmd.CommandType = CommandType.StoredProcedure;

if (rbNam.Checked)

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 1;

else

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 0;

cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = tbHoTen.Text.ToString();

cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = tbEmail.Text.ToString();

cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = tbSDT.Text.ToString();

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = dateTimePicker.Value.Date;

cmd.Parameters.Add("@Username", SqlDbType.VarChar).Value = tbTenDangNhap.Text.ToString();

cmd.Parameters.Add("@Password", SqlDbType.VarChar).Value = tbMatKhau.Text.ToString();

cmd.ExecuteNonQuery();

MessageBox.Show("Đã thêm thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

conn.Close();

this.Close();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.7.2 SỬA NHÂN VIÊN**

CREATE OR ALTER PROCEDURE SUA\_NHAN\_VIEN (@MaNhanVien varchar(10), @HoTen nvarchar(50), @GioiTinh bit,

@NgaySinh date, @SDT varchar(10), @Email varchar(50))

AS

BEGIN

BEGIN TRY

BEGIN TRAN

IF (DATEDIFF(year, @NgaySinh, GETDATE())<18)

BEGIN

RAISERROR(N'Độc giả chưa đủ 18 tuổi!!', 16,1)

RETURN

END

IF (len(@SDT)<>10)

BEGIN

RAISERROR(N'Số điện thoại không hợp lệ!!', 16,1)

RETURN

END

IF (@Email NOT LIKE '%\_\_\_@\_\_\_%.\_\_%')

BEGIN

RAISERROR(N'Email không hợp lệ!!', 16,1)

RETURN

END

UPDATE NHAN\_VIEN

SET HoTen = @HoTen, GioiTinh = @GioiTinh, NgaySinh = @NgaySinh, SoDienThoai = @SDT, Email = @Email

WHERE MaNhanVien = @MaNhanVien

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("SUA\_NHAN\_VIEN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaNhanVien", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.Parameters.Add("@HoTen", SqlDbType.NVarChar).Value = tbTen.Text.ToString();

if (cbGioiTinh.SelectedItem.ToString() == "Nữ")

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 0;

else

cmd.Parameters.Add("@GioiTinh", SqlDbType.Bit).Value = 1;

DateTime date = DateTime.ParseExact(tbNgaySinh.Text, "dd/MM/yyyy", null);

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = date;

cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = tbSDT.Text.ToString();

cmd.Parameters.Add("@Email", SqlDbType.VarChar).Value = tbEmail.Text.ToString();

cmd.ExecuteNonQuery();

MessageBox.Show("Thay đổi thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

conn.Close();

LoadData();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**-- 1.7.3 XOÁ NHÂN VIÊN**

CREATE OR ALTER PROCEDURE XOA\_NHAN\_VIEN (@MaNhanVien varchar(10))

AS

BEGIN

DELETE NHAN\_VIEN WHERE MaNhanVien = @MaNhanVien

END

GO

**// THỰC THI PROCEDURE**

SqlCommand cmd = new SqlCommand("XOA\_NHAN\_VIEN", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@MaNhanVien", SqlDbType.VarChar).Value = tbMa.Text.ToString();

cmd.ExecuteNonQuery();

### 1.8 Quản lý đăng nhập

**-- THAY ĐỔI MẬT KHẨU**

CREATE OR ALTER PROCEDURE DOI\_MAT\_KHAU (@TenDangNhap varchar(20), @MatKhauCu varchar(20), @MatKhauMoi varchar(20))

AS

BEGIN

BEGIN TRY

IF (len(@MatKhauMoi)<8)

BEGIN

RAISERROR(N'Mật khẩu quá ngắn!!', 16,1)

RETURN

END

IF (@MatKhauCu = (SELECT MatKhau FROM DANG\_NHAP WHERE TenDangNhap = @TenDangNhap))

BEGIN

UPDATE DANG\_NHAP

SET MatKhau = @MatKhauMoi

WHERE TenDangNhap = @TenDangNhap

DECLARE @sqlString varchar(MAX) = 'ALTER LOGIN [' + @TenDangNhap +'] WITH PASSWORD = N' + QUOTENAME(@MatKhauMoi,'''')

+ 'OLD\_PASSWORD = N' + QUOTENAME(@MatKhauCu,'''') + N';';

EXEC (@sqlString)

END

ELSE

BEGIN

RAISERROR(N'Sai mật khẩu!!', 16,1)

RETURN

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(MAX) = ERROR\_MESSAGE()

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY()

DECLARE @ErrorState INT = ERROR\_STATE()

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);

END CATCH

END

GO

**// THỰC THI PROCEDURE VÀ BẮT LỖI TỪ RAISERROR**

try

{

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("DOI\_MAT\_KHAU", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenDangNhap", SqlDbType.VarChar).Value = DbHelper.Username;

cmd.Parameters.Add("@MatKhauCu", SqlDbType.VarChar).Value = tbCu.Text;

cmd.Parameters.Add("@MatKhauMoi", SqlDbType.VarChar).Value = tbMoi.Text;

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

## Function

### 2.1 Tự động tạo mã

Các hàm tự động tạo mã này được lồng vào **default constraint** của các bảng để tự động sinh mã theo dạng, không được gọi trên C#

**-- 1.1 TỰ ĐỘNG TẠO MÃ SÁCH THEO DẠNG (VÍ DỤ: SACH00004)**

CREATE FUNCTION [Auto\_MaSach]()

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @MaSach VARCHAR(9)

IF (SELECT COUNT(\*) FROM SACH) = 0

SET @MaSach = '0'

ELSE

SELECT @MaSach = MAX(RIGHT(MaSach, 5)) FROM SACH

SELECT @MaSach = CASE

WHEN @MaSach >=0 AND @MaSach <9 THEN 'SACH0000' + CONVERT(CHAR, CONVERT(INT, @MaSach) + 1)

WHEN @MaSach >=9 AND @MaSach < 99 THEN 'SACH000' + CONVERT(CHAR, CONVERT(INT, @MaSach) + 1)

WHEN @MaSach >=99 AND @MaSach < 999 THEN 'SACH00' + CONVERT(CHAR, CONVERT(INT, @MaSach) + 1)

WHEN @MaSach >=999 THEN 'SACH0' + CONVERT(CHAR, CONVERT(INT, @MaSach) + 1)

END

RETURN @MaSach

END

GO

**-- 1.2 TỰ ĐỘNG TẠO MÃ PHIẾU MƯỢN THEO DẠNG (VÍ DỤ: 0000000010)**

CREATE FUNCTION [Auto\_MaPhieuMuon]()

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @MaPM VARCHAR(9)

IF (SELECT COUNT(\*) FROM PHIEU\_MUON) = 0

SET @MaPM = '0'

ELSE

SELECT @MaPM = MAX(RIGHT(MaPhieuMuon, 9)) FROM PHIEU\_MUON

SELECT @MaPM = CASE

WHEN @MaPM >=0 AND @MaPM <9 THEN '00000000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=9 AND @MaPM < 99 THEN '0000000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=99 AND @MaPM < 999 THEN '000000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=999 AND @MaPM < 9999 THEN '00000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=9999 AND @MaPM < 99999 THEN '0000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=99999 AND @MaPM < 999999 THEN '000' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=999999 AND @MaPM < 9999999 THEN '00' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

WHEN @MaPM >=9999999 AND @MaPM < 99999999 THEN '0' + CONVERT(CHAR, CONVERT(INT, @MaPM) + 1)

END

RETURN @MaPM

END

GO

**-- 1.3 TỰ ĐỘNG TẠO MÃ NHÂN VIÊN THEO DẠNG (VÍ DỤ: NV009)**

CREATE FUNCTION [dbo].[Auto\_MaNhanVien] ()

RETURNS VARCHAR(5)

AS

BEGIN

DECLARE @MaNV VARCHAR(10)

IF (SELECT COUNT(\*) FROM NHAN\_VIEN ) = 0

SET @MaNV = '0'

ELSE

SELECT @MaNV = MAX(RIGHT(MaNhanVien, 3)) FROM NHAN\_VIEN

SELECT @MaNV = CASE

WHEN @MaNV >= 0 and @MaNV <9 THEN 'NV00' + CONVERT(CHAR, CONVERT(INT,@MaNV) + 1)

WHEN @MaNV >= 9 THEN 'NV0' + CONVERT(CHAR, CONVERT(INT, @MaNV) +1 )

END

RETURN @MaNV

END

GO

**-- 1.4 TỰ ĐỘNG TẠO MÃ TÁC GIẢ THEO DẠNG (VÍ DỤ: TG0003)**

CREATE FUNCTION [dbo].[Auto\_MaTacGia] ()

RETURNS VARCHAR(6)

AS

BEGIN

DECLARE @MaTG VARCHAR(10)

IF (SELECT COUNT(\*) FROM TAC\_GIA ) = 0

SET @MaTG = '0'

ELSE

SELECT @MaTG = MAX(RIGHT(MaTacGia, 4)) FROM TAC\_GIA

SELECT @MaTG = CASE

WHEN @MaTG >= 0 and @MaTG <9 THEN 'TG000' + CONVERT(CHAR, CONVERT(INT,@MaTG) + 1)

WHEN @MaTG >= 9 and @MaTG < 99 THEN 'TG00' + CONVERT(CHAR, CONVERT(INT, @MaTG) +1 )

WHEN @MaTG >= 99 THEN 'TG0' + CONVERT(CHAR, CONVERT(INT, @MaTG) +1 )

END

RETURN @MaTG

END

GO

**-- 1.5 TỰ ĐỘNG TẠO MÃ NXB THEO DẠNG (VÍ DỤ: NXB0009)**

CREATE FUNCTION [dbo].[Auto\_MaNhaXuatBan] ()

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @MaNXB VARCHAR(7)

IF (SELECT COUNT(\*) FROM NHA\_XUAT\_BAN ) = 0

SET @MaNXB = '0'

ELSE

SELECT @MaNXB = MAX(RIGHT(MaNhaXuatBan, 4)) FROM NHA\_XUAT\_BAN

SELECT @MaNXB = CASE

WHEN @MaNXB >= 0 and @MaNXB <9 THEN 'NXB000' + CONVERT(CHAR, CONVERT(INT,@MaNXB) + 1)

WHEN @MaNXB >= 9 and @MaNXB < 99 THEN 'NXB00' + CONVERT(CHAR, CONVERT(INT, @MaNXB) +1 )

WHEN @MaNXB >= 99 THEN 'NXB0' + CONVERT(CHAR, CONVERT(INT, @MaNXB) +1 )

END

RETURN @MaNXB

END

GO

**-- 1.6 TỰ ĐỘNG TẠO MÃ CHUYÊN NGÀNH THEO DẠNG (VÍ DỤ: CN0004)**

CREATE FUNCTION [dbo].[Auto\_MaChuyenNganh] ()

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @MaCN VARCHAR(6)

IF (SELECT COUNT(\*) FROM CHUYEN\_NGANH ) = 0

SET @MaCN = '0'

ELSE

SELECT @MaCN = MAX(RIGHT(MaChuyenNganh, 4)) FROM CHUYEN\_NGANH

SELECT @MaCN = CASE

WHEN @MaCN >= 0 and @MaCN <9 THEN 'CN000' + CONVERT(CHAR, CONVERT(INT,@MaCN) + 1)

WHEN @MaCN >= 9 and @MaCN < 99 THEN 'CN00' + CONVERT(CHAR, CONVERT(INT, @MaCN) +1 )

WHEN @MaCN >= 99 THEN 'CN0' + CONVERT(CHAR, CONVERT(INT, @MaCN) +1 )

END

RETURN @MaCN

END

GO

### 2.2 Tìm kiếm

**-- 2.2.1 TÌM KIẾM ĐỘC GIẢ THEO MÃ ĐỘC GIẢ**

CREATE FUNCTION TIM\_KIEM\_MA\_DOC\_GIA (@MaDocGia varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_CHI\_TIET\_THE\_DOC\_GIA

WHERE [Mã độc giả] = @MaDocGia)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW**

SqlCommand cmd = new SqlCommand("SELECT \* FROM [dbo].[TIM\_KIEM\_MA\_DOC\_GIA] (@MaDocGia)", conn);

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@MaDocGia", tbTimKiemMaDocGia.Text);

dataAdapterDocGia = new SqlDataAdapter(cmd);

dataTableDocGia = new DataTable();

dataTableDocGia.Clear();

dataAdapterDocGia.Fill(dataTableDocGia);

dgvDocGia.DataSource = dataTableDocGia;

**-- 2.2.2 TÌM KIẾM SÁCH THEO TÊN**

CREATE FUNCTION TIM\_KIEM\_SACH (@TenSach nvarchar(100))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_CHI\_TIET\_SACH

WHERE [Tên sách] = @TenSach)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW**

SqlCommand cmd = new SqlCommand("SELECT \* FROM [dbo].[TIM\_KIEM\_SACH] (@TenSach)", conn);

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@TenSach", tbTimKiemTenSach.Text);

dataAdapterSach = new SqlDataAdapter(cmd);

dataTableSach = new DataTable();

dataTableSach.Clear();

dataAdapterSach.Fill(dataTableSach);

dgvSach.DataSource = dataTableSach;

**-- 2.2.3 TÌM KIẾM NHÂN VIÊN THEO TÊN**

CREATE FUNCTION TIM\_KIEM\_NHAN\_VIEN (@TenNhanVien nvarchar(50))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_CHI\_TIET\_NHAN\_VIEN

WHERE [Họ tên] = @TenNhanVien)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW**

SqlCommand cmd = new SqlCommand("SELECT \* FROM [dbo].[TIM\_KIEM\_NHAN\_VIEN] (@TenNhanVien)", conn);

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@TenNhanVien", tbTimKiemTen.Text);

dataAdapter = new SqlDataAdapter(cmd);

dataTable = new DataTable();

dataTable.Clear();

dataAdapter.Fill(dataTable);

dgv.DataSource = dataTable;

**-- 2.2.4 TÌM KIẾM CÁC SÁCH ĐANG MƯỢN THEO MÃ ĐỘC GIẢ**

CREATE FUNCTION TIM\_KIEM\_SACH\_DANG\_MUON\_MA\_DOC\_GIA (@MaDocGia varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_SACH\_DANG\_MUON

WHERE [Mã người mượn] = @MaDocGia)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

SqlConnection conn = DbHelper.Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("SELECT \* FROM [dbo].[TIM\_KIEM\_SACH\_DANG\_MUON\_MA\_DOC\_GIA] (@MaDocGia)", conn);

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@MaDocGia", tbTimKiemMaDocGia.Text);

dataAdapterMuonTra = new SqlDataAdapter(cmd);

dataTableMuonTra = new DataTable();

dataTableMuonTra.Clear();

dataAdapterMuonTra.Fill(dataTableMuonTra);

dgvMuonTra.DataSource = dataTableMuonTra;

**-- 2.2.5 TÌM KIẾM CÁC SÁCH ĐÃ TRẢ THEO MÃ ĐỘC GIẢ**

CREATE FUNCTION TIM\_KIEM\_SACH\_DA\_TRA\_MA\_DOC\_GIA (@MaDocGia varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_SACH\_DA\_TRA

WHERE [Mã người mượn] = @MaDocGia)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

**-- 2.2.6 TÌM KIẾM TOÀN BỘ MƯỢN TRẢ THEO MÃ ĐỘC GIẢ**

CREATE FUNCTION TIM\_KIEM\_MUON\_TRA\_MA\_DOC\_GIA (@MaDocGia varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_CHI\_TIET\_MUON\_TRA

WHERE [Mã người mượn] = @MaDocGia)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

**-- 2.2.7 TÌM KIẾM CÁC SÁCH ĐANG MƯỢN THEO MÃ PHIẾU MƯỢN**

CREATE FUNCTION TIM\_KIEM\_SACH\_DANG\_MUON\_PHIEU\_MUON (@MaPhieuMuon varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_SACH\_DANG\_MUON

WHERE [Mã phiếu mượn] = @MaPhieuMuon)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

**-- 2.2.8 TÌM KIẾM CÁC SÁCH ĐÃ TRẢ THEO MÃ PHIẾU MƯỢN**

CREATE FUNCTION TIM\_KIEM\_SACH\_DA\_TRA\_PHIEU\_MUON(@MaPhieuMuon varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_SACH\_DA\_TRA

WHERE [Mã phiếu mượn] = @MaPhieuMuon)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

**-- 2.2.9 TÌM KIẾM CÁC SÁCH ĐÃ TRẢ THEO MÃ PHIẾU MƯỢN**

CREATE FUNCTION TIM\_KIEM\_MUON\_TRA\_PHIEU\_MUON (@MaPhieuMuon varchar(10))

RETURNS TABLE

AS RETURN (

SELECT \* FROM VIEW\_CHI\_TIET\_MUON\_TRA

WHERE [Mã phiếu mượn] = @MaPhieuMuon)

GO

**// SỬ DỤNG FUNCTION ĐỂ HIỂN THỊ KẾT QUẢ LÊN DATAGRIDVIEW: TƯƠNG TỰ VỚI CÁC HÀM TÌM KIẾM KHÁC**

### 2.3 Lấy mã nhân viên khi đăng nhập

**-- LẤY MÃ NHÂN VIÊN SAU KHI ĐĂNG NHẬP (SỬ DỤNG MÃ NHÂN VIÊN CHO PHIÊN ĐĂNG NHẬP ĐÓ)**

CREATE OR ALTER FUNCTION LAY\_MA\_NHAN\_VIEN (@Username varchar(20))

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @MaNhanVien varchar(10)

SET @MaNhanVien= (SELECT MaNhanVien FROM DANG\_NHAP WHERE TenDangNhap = @Username)

RETURN @MaNhanVien

END

GO

**// SỬ DỤNG FUNCTION**

SqlCommand cmd = new SqlCommand("LAY\_MA\_NHAN\_VIEN", conn);

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT [dbo].[LAY\_MA\_NHAN\_VIEN] (@Username)";

cmd.Parameters.AddWithValue("@Username", DbHelper.Username);

object staffId = cmd.ExecuteScalar();

DbHelper.MaNhanVien = staffId.ToString();

### 2.4 Tính tiền phạt

**-- 2.4.1 TÍNH TIỀN PHẠT THEO TÌNH TRẠNG KHI TRẢ SÁCH**

CREATE FUNCTION TINH\_PHAT\_HU\_HONG (@MaSach varchar(10), @TinhTrang int)

RETURNS DECIMAL

AS

BEGIN

DECLARE @GiaBia decimal = (SELECT GiaBia FROM SACH WHERE MaSach = @MaSach)

IF (@TinhTrang = 2)

RETURN (@GiaBia \* 5)

RETURN 0

END

GO

**// SỬ DỤNG FUNCTION ĐỂ TÍNH TIỀN TRỄ HẠN VÀ GÁN VÀO BIẾN KẾT QUẢ**

SqlCommand cmd = new SqlCommand("SELECT [DBO].[TINH\_PHAT\_TRE\_HAN](@NgayHetHan)", conn);

cmd.CommandType = CommandType.Text;

DateTime date = DateTime.ParseExact(tbNgayHetHan.Text, "dd/MM/yyyy", null);

cmd.Parameters.Add("@NgayHetHan", SqlDbType.Date).Value = date;

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

phatQuaHan = dr.GetDecimal(0);

**-- 2.4.2 TÍNH TIỀN PHẠT TRỄ HẠN KHI TRẢ SÁCH**

CREATE OR ALTER FUNCTION TINH\_PHAT\_TRE\_HAN (@NgayHetHan date)

RETURNS DECIMAL

AS

BEGIN

IF (GETDATE() > @NgayHetHan)

RETURN 1000 \* DATEDIFF(dd, @NgayHetHan, GETDATE())

RETURN 0

END

GO

**// SỬ DỤNG FUNCTION ĐỂ TÍNH TIỀN HƯ HỎNG VÀ GÁN VÀO BIẾN KẾT QUẢ**

SqlCommand cmd = new SqlCommand("SELECT [DBO].[TINH\_PHAT\_HU\_HONG](@MaSach, @TinhTrang)", conn);

cmd.CommandType = CommandType.Text;

cmd.Parameters.Add("@MaSach", SqlDbType.VarChar).Value = FormMuonTraSach.maSach;

cmd.Parameters.Add("@TinhTrang", SqlDbType.Int).Value = cbTinhTrang.SelectedValue;

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

phatHuHong = dr.GetDecimal(0);

# CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN

Phân quyền ứng với hai nhóm người dùng:

## 1. Nhân viên

Role NhanVien dành cho nhóm người dùng có vai trò là thủ thư, thực hiện các việc quản lý nhập liệu, cho mượn trả sách trong thư viện  
Tạo mới và gán các quyền cần thiết cho Role NhanVien:

CREATE ROLE NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON CHI\_TIET\_MUON\_TRA TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON CHUYEN\_NGANH TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON DOC\_GIA TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON NHA\_XUAT\_BAN TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON PHIEU\_MUON TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON SACH TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON SACH\_TAC\_GIA TO NhanVien

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON TAC\_GIA TO NhanVien

GRANT SELECT, REFERENCES ON DOI\_TUONG\_DOC\_GIA TO NhanVien

GRANT SELECT, REFERENCES ON TINH\_TRANG\_SACH TO NhanVien

GRANT SELECT, UPDATE, REFERENCES ON DANG\_NHAP TO NhanVien

GRANT SELECT, REFERENCES ON NHAN\_VIEN TO NhanVien

GRANT EXECUTE TO NhanVien

GRANT SELECT TO NhanVien

DENY UPDATE, INSERT, DELETE ON NHAN\_VIEN to NhanVien

DENY EXECUTE ON SUA\_NHAN\_VIEN to NhanVien

DENY EXECUTE ON THEM\_NHAN\_VIEN to NhanVien

DENY EXECUTE ON XOA\_NHAN\_VIEN to NhanVien

GO

## 2. Quản trị viên

Đối với Quản trị viên, yêu cầu cần có quyền tạo login cho các tài khoản nhân viên nên ta sử dụng server sole *sysadmin*

## 3. Trigger tạo User

Trigger tạo login cho user khi tài khoản được tạo trong table [DANG\_NHAP] và gán user vào role tương ứng sau khi tài khoản đăng nhập được tạo:

CREATE OR ALTER TRIGGER TRIGGER\_SQL\_ACCOUNT ON DANG\_NHAP

AFTER INSERT

AS

DECLARE @userName nvarchar(20), @passWord nvarchar(20)

SELECT @userName=i.TenDangNhap, @passWord=i.MatKhau

FROM inserted i

BEGIN

DECLARE @sqlString nvarchar(2000)

SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+ @passWord +''',

DEFAULT\_DATABASE=[QuanLyThuVien], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName

EXEC (@sqlString)

DECLARE @role bit

SELECT @role = i.LoaiTaiKhoan FROM inserted i

IF (@role = 0)

SET @sqlString = 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @userName;

ELSE

SET @sqlString = 'ALTER ROLE NhanVien ADD MEMBER ' + @userName;

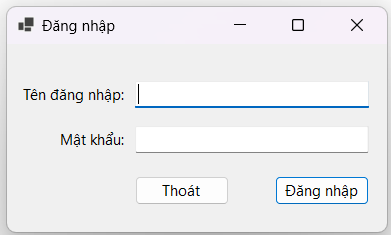
EXEC (@sqlString)

END

GO

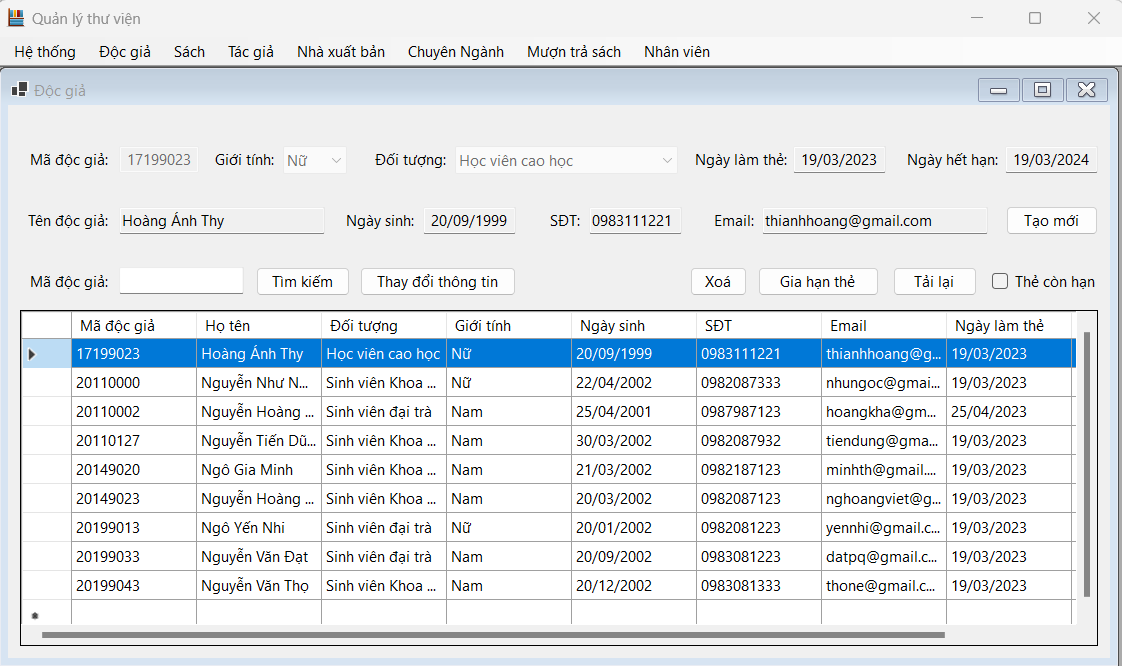
# CHƯƠNG 5. THIẾT KẾ GIAO DIỆN HỆ THỐNG

## 1. Đăng nhập



## 2. Độc giả

### 2.1 Form độc giả



- Hiển thị dữ liệu Độc giả:

SqlConnection conn = DbHelper.Connect();

conn.Open();

if (checkBoxConHan.Checked == false)

dataAdapterDocGia = new SqlDataAdapter("SELECT \* FROM VIEW\_CHI\_TIET\_THE\_DOC\_GIA", conn);

else

dataAdapterDocGia = new SqlDataAdapter("SELECT \* FROM VIEW\_DOC\_GIA\_CON\_HAN", conn);

dataTableDocGia = new DataTable();

dataTableDocGia.Clear();

dataAdapterDocGia.Fill(dataTableDocGia);

dgvDocGia.DataSource = dataTableDocGia;

dataAdapterDoiTuong = new SqlDataAdapter("SELECT \* FROM DOI\_TUONG\_DOC\_GIA", conn);

dataTableDoiTuong = new DataTable();

dataTableDoiTuong.Clear();

dataAdapterDoiTuong.Fill(dataTableDoiTuong);

cbDoiTuong.DataSource = dataTableDoiTuong;

cbDoiTuong.DisplayMember = "TenDoiTuong";

cbDoiTuong.ValueMember = "MaDoiTuong";

cbDoiTuong.SelectedIndex = -1;

cbGioiTinh.Items.Add("Nữ");

cbGioiTinh.Items.Add("Nam");

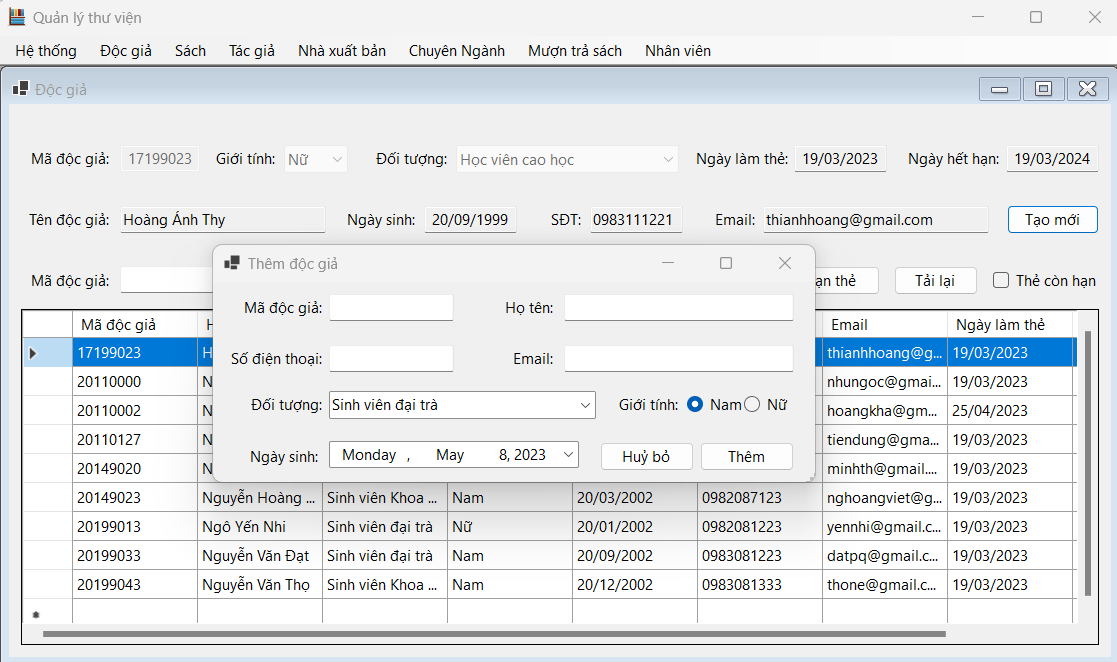
dgvDocGia.Columns[4].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvDocGia.Columns[7].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvDocGia.Columns[8].DefaultCellStyle.Format = "dd/MM/yyyy";

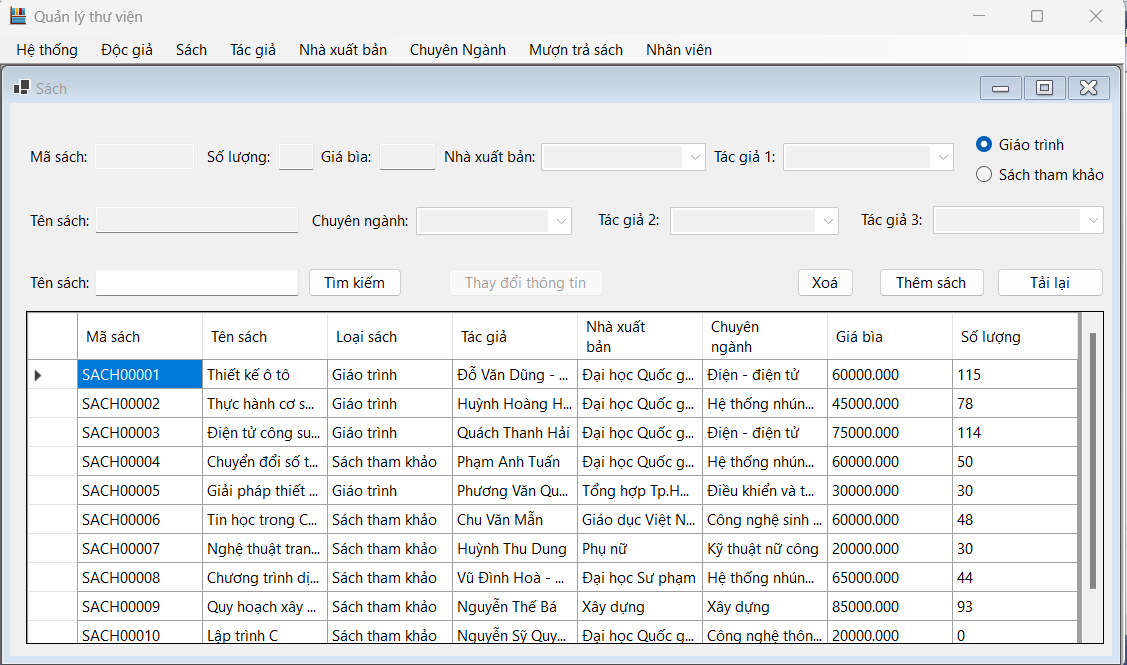
### 2.2 Dialog Thêm độc giả

- Xuất hiện khi chọn nút [Tạo mới] trong Form Độc Giả



## 3. Sách

### 3.1 Form Sách



- Hiển thị dữ liệu Sách:

SqlConnection conn = DbHelper.Connect();

conn.Open();

dataAdapterSach = new SqlDataAdapter("SELECT \* FROM VIEW\_CHI\_TIET\_SACH", conn);

dataTableSach = new DataTable();

dataTableSach.Clear();

dataAdapterSach.Fill(dataTableSach);

dgvSach.DataSource = dataTableSach;

dataAdapterNhaXuatBan = new SqlDataAdapter("SELECT \* FROM NHA\_XUAT\_BAN", conn);

dataTableNhaXuatBan = new DataTable();

dataAdapterNhaXuatBan.Fill(dataTableNhaXuatBan);

cbNhaXuatBan.DataSource = dataTableNhaXuatBan;

cbNhaXuatBan.DisplayMember = "TenNhaXuatBan";

cbNhaXuatBan.ValueMember = "MaNhaXuatBan";

dataAdapterChuyenNganh = new SqlDataAdapter("SELECT \* FROM CHUYEN\_NGANH", conn);

dataTableChuyenNganh = new DataTable();

dataAdapterChuyenNganh.Fill(dataTableChuyenNganh);

cbChuyenNganh.DataSource = dataTableChuyenNganh;

cbChuyenNganh.DisplayMember = "TenChuyenNganh";

cbChuyenNganh.ValueMember = "MaChuyenNganh";

dataAdapterTacGia = new SqlDataAdapter("SELECT \* FROM TAC\_GIA", conn);

dataTableTacGia1 = new DataTable();

dataAdapterTacGia.Fill(dataTableTacGia1);

cbTacGia1.DataSource = dataTableTacGia1;

cbTacGia1.DisplayMember = "TenTacGia";

cbTacGia1.ValueMember = "MaTacGia";

dataTableTacGia2 = new DataTable();

dataAdapterTacGia.Fill(dataTableTacGia2);

cbTacGia2.DataSource = dataTableTacGia2;

cbTacGia2.DisplayMember = "TenTacGia";

cbTacGia2.ValueMember = "MaTacGia";

cbTacGia2.SelectedIndex = -1;

dataTableTacGia3 = new DataTable();

dataAdapterTacGia.Fill(dataTableTacGia3);

cbTacGia3.DataSource = dataTableTacGia3;

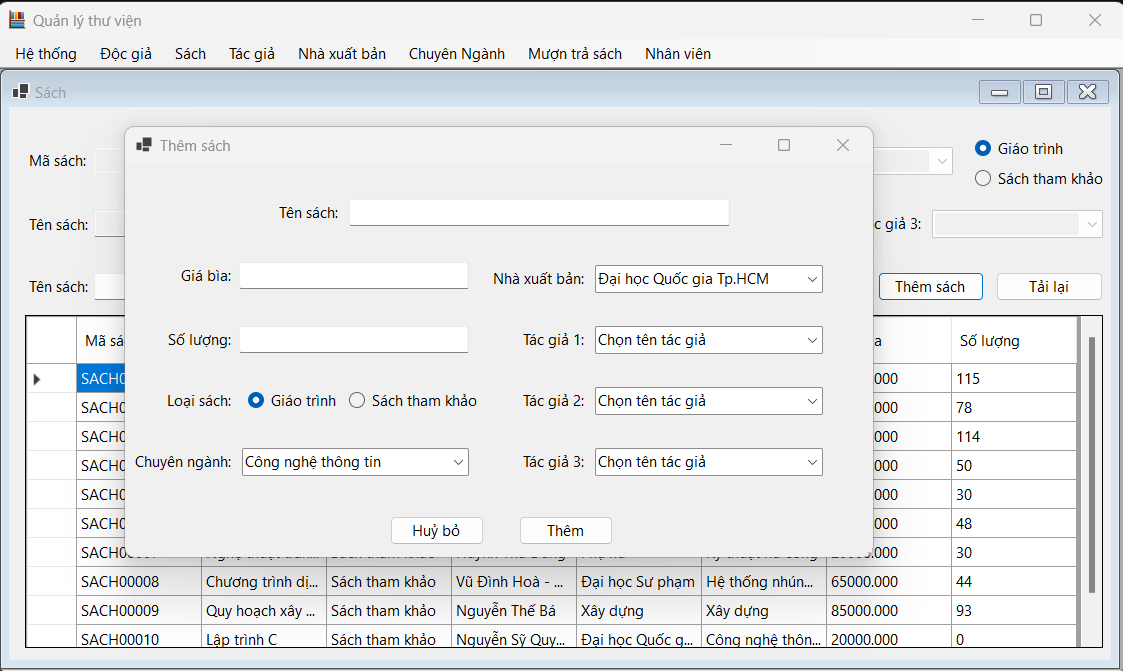
cbTacGia3.DisplayMember = "TenTacGia";

cbTacGia3.ValueMember = "MaTacGia";

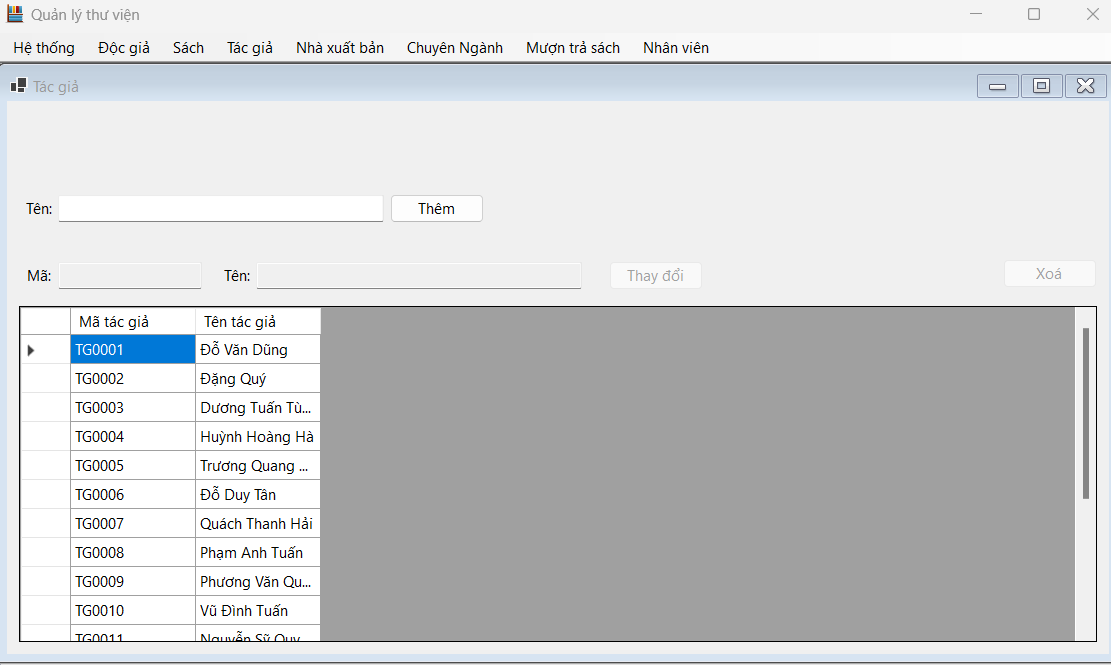
cbTacGia3.SelectedIndex = -1;

### 3.2 Dialog Thêm sách

- Xuất hiện khi chọn nút [Thêm sách] trong Form Sách



## 4. Tác giả



- Hiển thị dữ liệu Tác giả:

SqlConnection conn = DbHelper.Connect();

conn.Open();

dataAdapterTacGia = new SqlDataAdapter("SELECT \* FROM VIEW\_TAC\_GIA", conn);

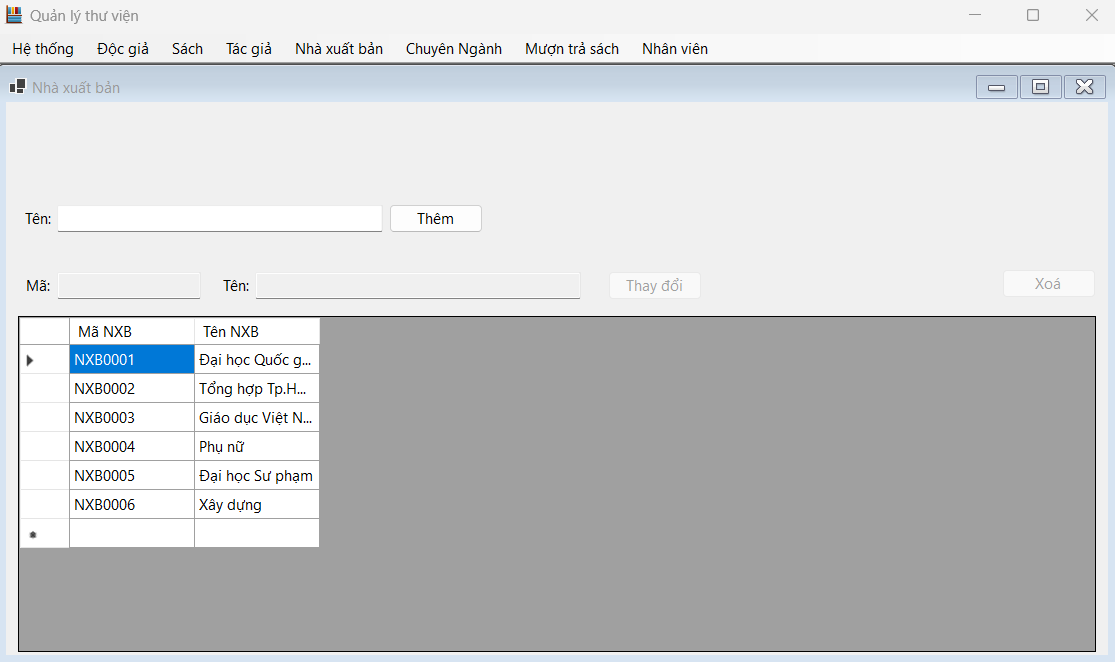
dataTableTacGia = new DataTable();

dataTableTacGia.Clear();

dataAdapterTacGia.Fill(dataTableTacGia);

dgv.DataSource = dataTableTacGia;

## 5. Nhà xuất bản



- Hiển thị dữ liệu Nhà xuất bản:

SqlConnection conn = DbHelper.Connect();

conn.Open();

dataAdapterNXB = new SqlDataAdapter("SELECT \* FROM VIEW\_NHA\_XUAT\_BAN", conn);

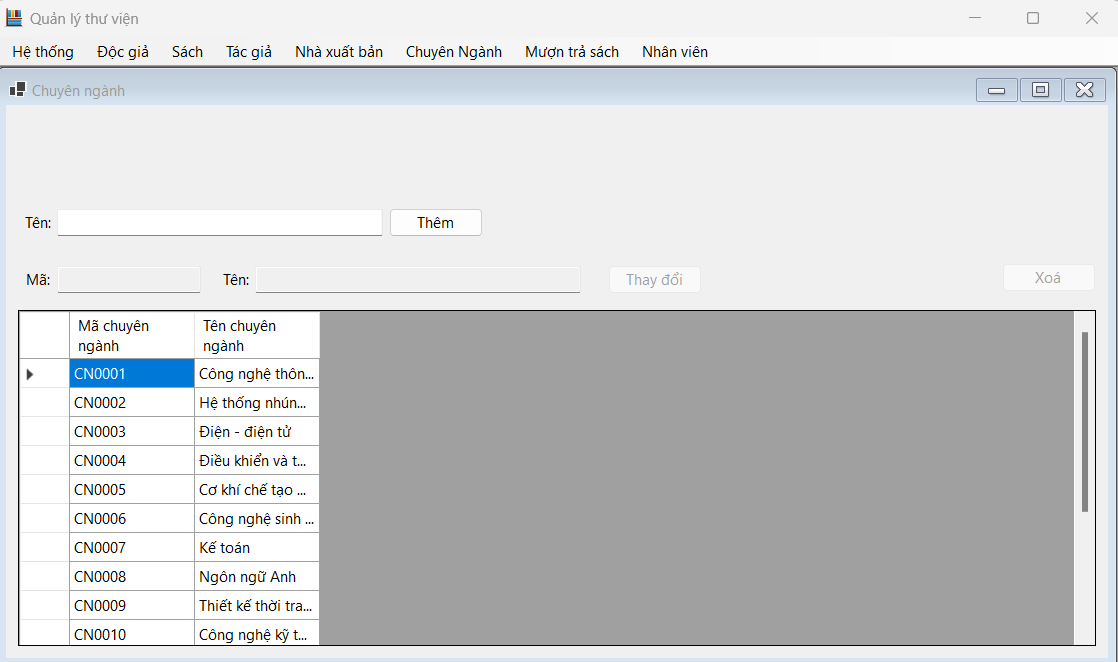
dataTableNXB = new DataTable();

dataTableNXB.Clear();

dataAdapterNXB.Fill(dataTableNXB);

dgv.DataSource = dataTableNXB;

## 6. Chuyên ngành



- Hiển thị dữ liệu Chuyên ngành:

SqlConnection conn = DbHelper.Connect();

conn.Open();

dataAdapterChuyenNganh = new SqlDataAdapter("SELECT \* FROM VIEW\_CHUYEN\_NGANH", conn);

dataTableChuyenNganh = new DataTable();

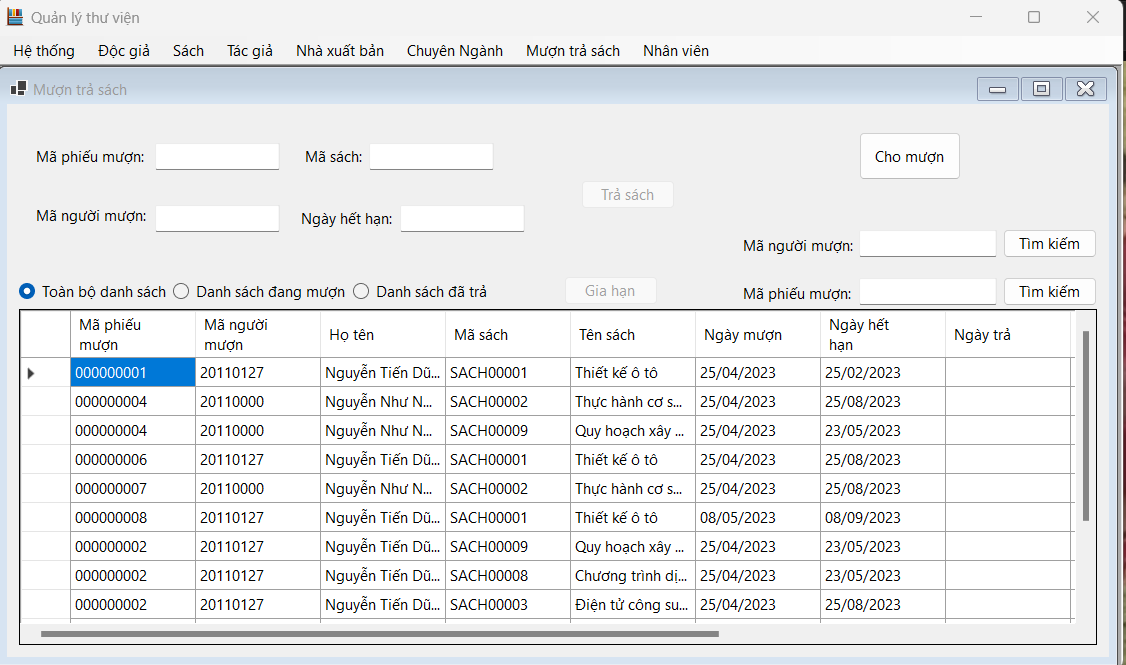
dataTableChuyenNganh.Clear();

dataAdapterChuyenNganh.Fill(dataTableChuyenNganh);

dgv.DataSource = dataTableChuyenNganh;

## 7. Mượn trả sách

### 7.1 Form Mượn trả sách



- Hiển thị dữ liệu mượn trả sách:

SqlConnection conn = DbHelper.Connect();

conn.Open();

if (rbToanBo.Checked)

{

dataAdapterMuonTra = new SqlDataAdapter("SELECT \* FROM VIEW\_CHI\_TIET\_MUON\_TRA", conn);

dataTableMuonTra = new DataTable();

dataTableMuonTra.Clear();

dataAdapterMuonTra.Fill(dataTableMuonTra);

dgvMuonTra.DataSource = dataTableMuonTra;

dgvMuonTra.Columns[5].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvMuonTra.Columns[6].DefaultCellStyle.Format = "dd/MM/yyyy";

}

else if (rbDangMuon.Checked)

{

dataAdapterMuonTra = new SqlDataAdapter("SELECT \* FROM VIEW\_SACH\_DANG\_MUON", conn);

dataTableMuonTra = new DataTable();

dataTableMuonTra.Clear();

dataAdapterMuonTra.Fill(dataTableMuonTra);

dgvMuonTra.DataSource = dataTableMuonTra;

dgvMuonTra.Columns[5].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvMuonTra.Columns[6].DefaultCellStyle.Format = "dd/MM/yyyy";

}

else if (rbDaTra.Checked)

{

dataAdapterMuonTra = new SqlDataAdapter("SELECT \* FROM VIEW\_SACH\_DA\_TRA", conn);

dataTableMuonTra = new DataTable();

dataTableMuonTra.Clear();

dataAdapterMuonTra.Fill(dataTableMuonTra);

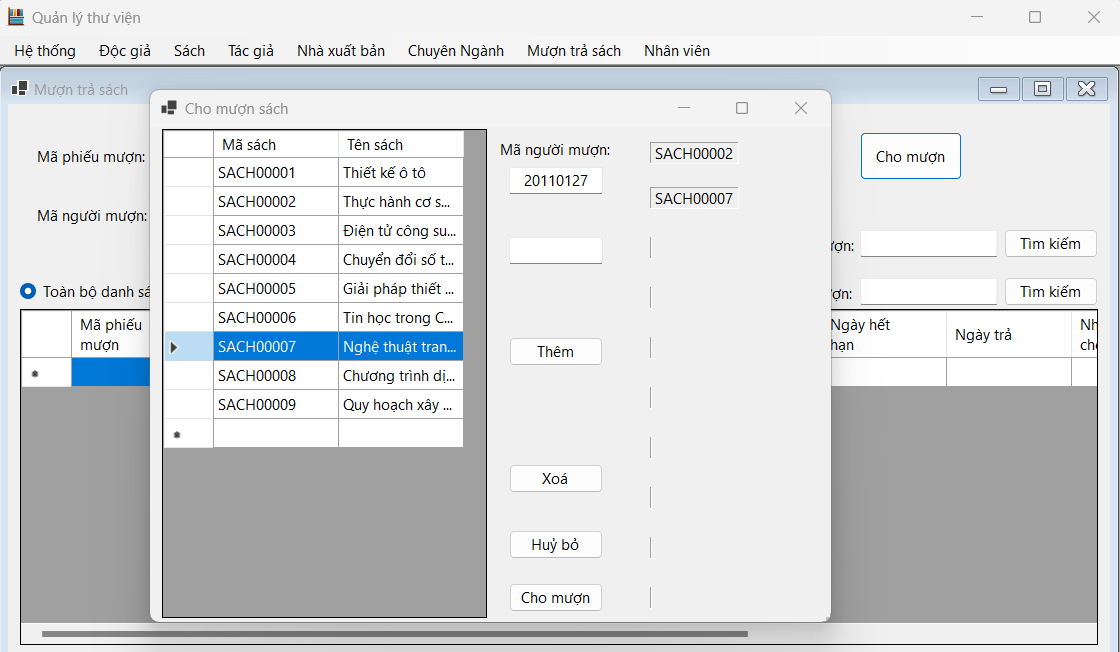
dgvMuonTra.DataSource = dataTableMuonTra;

dgvMuonTra.Columns[5].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvMuonTra.Columns[6].DefaultCellStyle.Format = "dd/MM/yyyy";

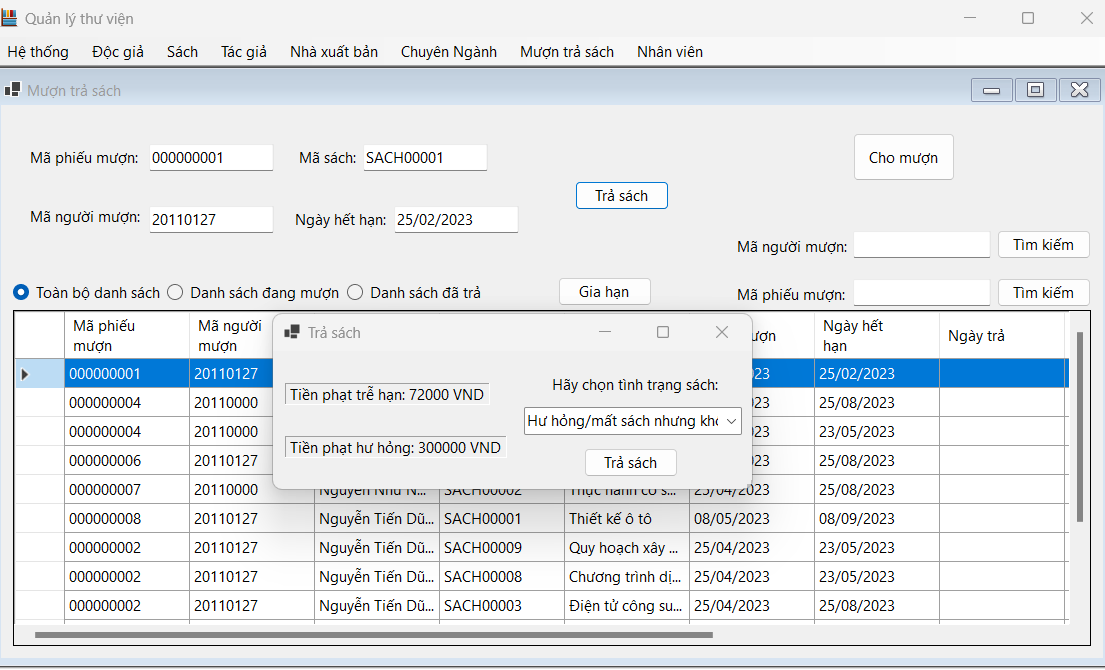
### 7.2 Dialog cho mượn sách

Xuất hiện khi chọn nút [Cho mượn] trong Form Mượn trả sách



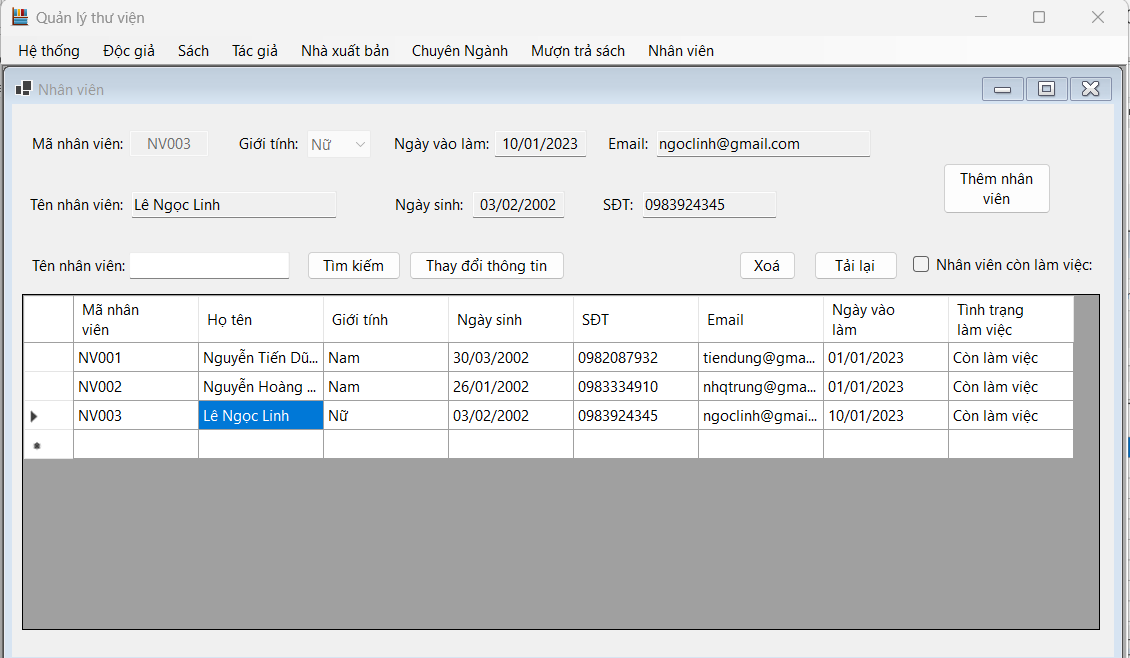
### 7.3 Dialog Trả sách

Xuất hiện khi chọn nút [Trả sách] trong Form Mượn trả sách



## 8. Nhân viên

### 8.1 Form Nhân viên



- Hiển thị dữ liệu Nhân viên:

SqlConnection conn = DbHelper.Connect();

conn.Open();

if (checkBoxConLamViec.Checked == false)

dataAdapter = new SqlDataAdapter("SELECT \* FROM VIEW\_CHI\_TIET\_NHAN\_VIEN", conn);

else

dataAdapter = new SqlDataAdapter("SELECT \* FROM VIEW\_NHAN\_VIEN\_LAM\_VIEC", conn);

dataTable = new DataTable();

dataTable.Clear();

dataAdapter.Fill(dataTable);

dgv.DataSource = dataTable;

cbGioiTinh.Items.Add("Nữ");

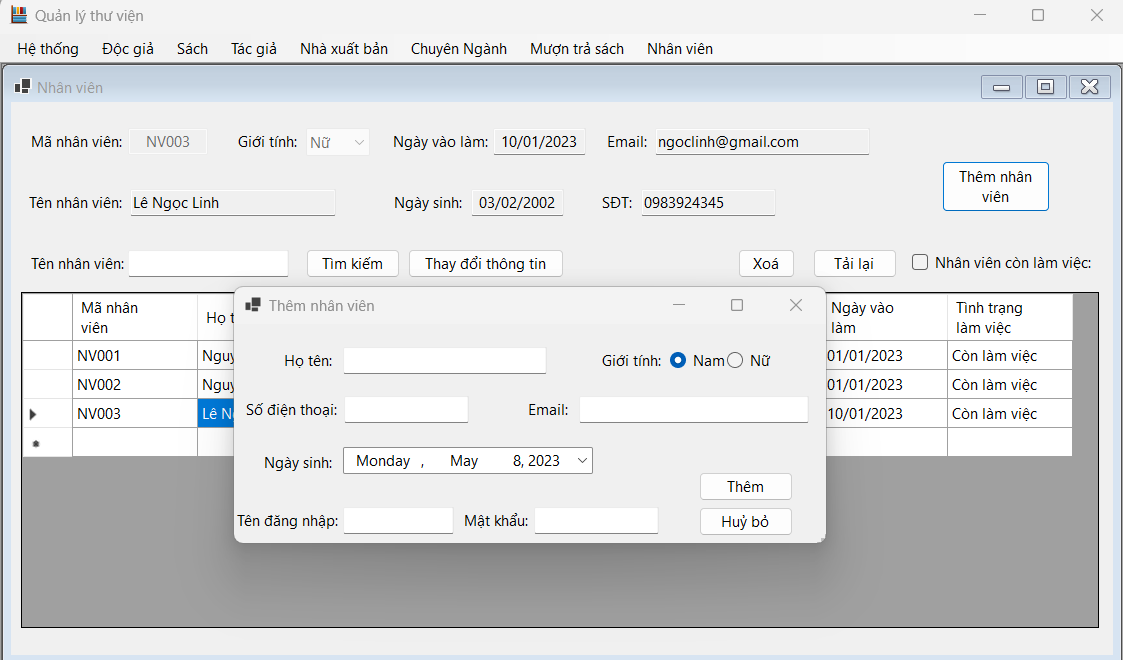
cbGioiTinh.Items.Add("Nam");

dgv.Columns[3].DefaultCellStyle.Format = "dd/MM/yyyy";

dgv.Columns[6].DefaultCellStyle.Format = "dd/MM/yyyy";

### 8.2 Dialog Thêm nhân viên

Xuất hiện khi chọn nút [Thêm nhân viên] trong Form Nhân viên



## 9. Đổi mật khẩu

