**ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

-----🙞🙜🕮🙞🙜-----

**NGÀNH CÔNG NGHỆ THÔNG TIN**

BÁO CÁO ĐỒ ÁN HỆ QUẢN TRỊ

CƠ SỞ DỮ LIỆU

**ĐỀ TÀI: QUẢN LÝ THƯ VIỆN**

**TRONG TRƯỜNG ĐẠI HỌC**

**Nhóm : 4**

Nguyễn Đức Khải 20110655

Lê Nguyễn Tuấn Kiệt 20110286

Nguyễn Đức Thịnh 20110256

Hà Vĩ Khang 20110656

**GVHD: Ths.Nguyễn Thành Sơn**

**TP.Hồ Chí Minh, tháng 9 năm 2022**

**Mục Lục**

[1. Giới thiệu 3](#_Toc116157508)

[1.1. Lý do chọn đề tài 3](#_Toc116157509)

[1.2. Mục tiêu 4](#_Toc116157510)

[2. Mô tả hệ thống 5](#_Toc116157511)

[2.1. Dữ liệu cần lưu 5](#_Toc116157512)

[2.1.1 Đầu sách 5](#_Toc116157513)

[2.1.2 Sách 5](#_Toc116157514)

[2.1.3 Danh mục 5](#_Toc116157515)

[2.1.4 Ngôn ngữ 5](#_Toc116157516)

[2.1.5 Nhà xuất bản 5](#_Toc116157517)

[2.1.6 Tác Giả 6](#_Toc116157518)

[2.1.7 Tài khoản 6](#_Toc116157519)

[2.1.8 Hồ sơ người sử dụng 6](#_Toc116157520)

[2.2. Đặc tả 6](#_Toc116157521)

[2.3. Chức năng 7](#_Toc116157522)

[3. Thiết kế cơ sở dữ liệu 7](#_Toc116157523)

[3.1. Thiết kế mức logic 8](#_Toc116157524)

[3.2. Mô tả các thuộc tính và kiểu dữ liệu của chúng 8](#_Toc116157525)

[4. Cài đặt cơ sở dữ liệu và ràng buộc 13](#_Toc116157526)

[4.1. Tạo bảng TheLoai: 14](#_Toc116157527)

[4.2. Tạo bảng NgonNgu 14](#_Toc116157528)

[4.3. Tạo bảng TacGia 14](#_Toc116157529)

[4.4. Tạo bảng NhaXuanBan 14](#_Toc116157530)

[4.5. Tạo bảng VaiTro 15](#_Toc116157531)

[4.6. Tạo bảng DocGia 15](#_Toc116157532)

[4.7. Tạo bảng DauSach 15](#_Toc116157533)

[4.8. Tạo bảng HoSo 16](#_Toc116157534)

[4.9. Tạo bảng NhanVien 17](#_Toc116157535)

[4.10. Tạo bảng Muon 17](#_Toc116157536)

[4.11. Tạo bảng tacgia\_sach 18](#_Toc116157537)

[4.12. Tạo bảng vaitro\_nhanvien 18](#_Toc116157538)

[4.13. Tạo bảng Sach 19](#_Toc116157539)

[4.14. Tạo bảng MuonSach 19](#_Toc116157540)

[5. Cài đặt các chức năng 19](#_Toc116157541)

[5.1. View 19](#_Toc116157542)

[5.2. Trigger 20](#_Toc116157543)

[5.3. Stored procedure 22](#_Toc116157544)

[5.4. Function 22](#_Toc116157545)

# Giới thiệu

## Lý do chọn đề tài

Ở bất kỳ thời kỳ lịch sử nào, “Thư viện” đều được coi là kho trí tuệ của loài người. Quy mô thư viện gắn liền với sự phát triển của loài người, cùng với sự phát triển của xã hội và phân ngành sản xuất, thư viện ngày nay ngày càng đa dạng về nội dung và lớn về số lượng.

Nhu cầu sử dụng thư viện rất rộng rãi. Có lẽ, không có một lĩnh vực hoạt động nào trong xã hội không cần đến thư viện, đặc biệt là trong môi trường giáo dục ở các trường đại học.

Tuy nhiên hiện nay, việc vận dụng ngay các phần mềm chuyên dụng còn là một vấn đề gặp không ít khó khăn, các hệ thống tin học hóa chưa đáp ứng được nhu cầu của các nhà quản lý, các quá trình tìm kiếm, sử dụng vẫn còn những thao tác thủ công, bất tiện, không phù hợp với nhu cầu người sử dụng.

Ngày nay, nhiều thư viện đã có số lượng hàng trăm cuốn sách, số lượng lớn sinh viên đến mượn sách hàng ngày. Điều đó tạo nên sự sức tạp về sự quản lý và tra tìm sách. Cùng với sự chuyển đổi số, nhóm chúng em nảy ra ý tưởng tạo ra các phần mềm để hỗ trợ quá trình quản lý sách trong thư viện một cách hiệu quả.

Vì vậy, nhóm chúng em làm đề tài **“Quản lý thư viện trong trường đại học“.**

## Mục tiêu

Thông qua đồ án “ **Quản lý thư viện trong trường đại học** “**,** nhóm em muốn tích hợp các kiến thức đã học ở môn **“ Hệ quản trị cơ sở dữ liệu “** vào thực tiễn. Luyện tập, rèn luyện và nâng cao kiến thực, cũng như kỹ năng làm việc nhóm của bản thân mình.

# Mô tả hệ thống

## Dữ liệu cần lưu

### Đầu sách

+ Tên sách

+ Giá sách

+ Số lượng sách

+ Mô tả sách

+ Số trang

+ Hình ảnh

+ Loại sách

+ Ngày xuất bản

+ Mã ngôn ngữ

+ Mã danh mục

+ Mã tác giả

+ Mã nhà xuất bản

### Sách

+ Trạng thái

+ Vị trí

+ Mã đầu sách

+ Mã giao dịch

### Danh mục

+ Tên danh mục

+ Mã danh mục

### Ngôn ngữ

+ Tên ngôn ngữ

+ Mã ngôn ngữ

### Nhà xuất bản

**+** Tên nhà xuất bản

+ Mã nhà xuất bản

### Tác Giả

+ Tên

+ Mã tác giả

### Tài khoản

+ Tên đăng nhập

+ Email

+ Mật khẩu

+ Trạng thái

+ Mã hồ sơ

### Hồ sơ người sử dụng

+ Họ

+ Tên chính

+ Địa chỉ

+ Số điện thoại

+ Hình ảnh

## Đặc tả

Một trường đại học xây dựng mô hình quản lý việc mượn, trả sách trong thư viện bao gồm: Các **đầu sách** với các thông tin như mã đầu sách, tiêu đề sách, mô tả sách, giá bán, ngày xuất bản, hình ảnh, loại ( sách giao khoa hoặc sách tham khảo), trạng thái, nhà xuất bản, ngôn ngữ, thể loại.

Một đầu sách có duy nhất một **nhà xuất bản** gồm: mã nhà xuất bản, tên nhà xuất bản. Một nhà xuất bản có thể xuất bản một hoặc nhiều đầu sách

Một đầu sách có thể có một hoặc nhiều **tác giả** gồm: mã tác giả, tên tác giả. Một tác giả có thể sở hữu một hoặc nhiều đầu sách

Một đầu sách có duy nhất một **ngôn ngữ** bao gồm mã ngôn ngữ và tên ngôn ngữ, nếu đầu sách được viết bằng nhiều ngôn ngữ, thì sẽ được gọi là “song ngữ”. Một ngôn ngữ có thể không có hoặc có nhiều đầu sách

Một đầu sách chỉ thuộc về duy nhất một **thể loại** bao gồm: mã thể loại và tên thể loại. Một thể loại có thể có một hoặc nhiều đầu sách

**Cuốn sách** là các cá thể sách riêng biệt của một đầu sách gồm mã đầu sách, mã sách, vị trí, trạng thái.

Về phần quản lý hệ thống cung cấp các tài khoản cho **nhân viên** bao gồm mã tài khoản, tên đăng nhập, Email, mật khẩu, trạng thái ( còn hoạt động, không còn hoạt động ).

Mỗi nhân viên còn có **hồ sơ** gồm các thông tin: tên, họ, địa chỉ, số điện thoại, hình ảnh, email, giới tính, ngày sinh, lương.

Môi nhân viên được phân quyền với **vai trò**: thủ thư, quản lý. Một nhân viên có thể có một hoặc nhiều vai trò. Một vai trò có thể được trao cho một hoặc nhiều nhân viên

Sinh viên, giảng viên đến mượn sách còn được gọi là **độc giả,** bao gồm các thuộc tính: mã độc giả, tên độc giả, mã số sinh viên, khoa, giới tính, trạng thái ( cho phép mượn, không được phép mượn), ngày sinh, số điện thoại, email, hình ảnh.

Việc **mượn** sách sẽ ghi tại các thông tin gồm mã giao dịch, ngày giao dịch, ngày trả, ngày hết hạn, mã độc giả mượn và mã nhân viên cho mượn sách, ngoài ra nếu người mượn không trả sách đúng hạn sẽ phải trả thêm phí phạt dựa theo thời gian quá hạn. Một độc giả và nhân viên có thể không có hoặc có tham gia vào nhiều phiếu mượn, một phiếu mượn chỉ thuộc về duy nhất một độc giả và duy nhất một nhân viên. Một phiếu mượn bao gồm một hoặc nhiều đầu sách. Một đầu sách có thể không tham gia hoặc tham gia vào nhiều phiếu mượn

Một phiếu mượn bao gồm một hoặc nhiều đầu sách, được gọi là **mượn sách.** Bao gồmmã sách, mã phiếu mượn, ghi chú, trạng thái ( đang mượn, đã trả )

## Chức năng

Đối với người quản lý, cung cấp những tính năng để quản lý dữ liệu như:

+ Quản lý đầu sách

+ Quản lý nhân viên và tài khoản nhân viên

+ Quản lý phiếu mượn

+ Quản lý độc giả

Đối với thủ thư, cung cấp những tính năng để tương tác với độc giả và quản lý việc mượn/trả sách như

+ Quản lý đầu sách

+ Quản lý phiếu mượn

+ Quản lý độc giả

+ Tạo phiếu mượn

# Thiết kế cơ sở dữ liệu

## Mô tả các thuộc tính và kiểu dữ liệu của chúng

**Bảng 1. TheLoai**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã thể loại | Khóa chính |
| 2 | ten | nvarchar(50) | Tên thể loai | Not null  Check(ten NOT LIKE '%[^a-zA-Z]%') |

**Bảng 2. NgonNgu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã ngôn ngũ | Khóa chính |
| 2 | ten | nvarchar(20) | Tên ngôn ngữ | Not null  Check(ten NOT LIKE '%[^a-zA-Z ]%') |

**Bảng 3. Tacgia**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã tác giả | Khóa chính |
| 2 | ten | nvarchar(50) | Tên tác giả | Not null  Check(ten NOT LIKE '%[^a-zA-Z ]%') |

**Bảng 4. NhaXuatBan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã nhà xuất bản | Khóa chính |
| 2 | ten | nvarchar(50) | Tên nhà xuất bản | Not null |

**Bảng 5. VaiTro**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã vai trò | Khóa chính |
| 2 | ten | nvarchar(20) | Tên vai trò | Not null |
| 3 | mota | nvarchar(200) | Mô tả vai trò |  |

**Bảng 6. DocGia**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã độc giả | Khóa chính |
| 2 | ten | nvarchar(50) | Tên độc giả | Not null |
| 3 | mssv | nvarchar(10) | Mã số sinh viên | Not null |
| 4 | Khoa | nvarchar(50) | Tên khoa | Not null |
| 5 | trangthai | int | Trạng thái của độc giả | Not null |
| 6 | ngaysinh | datetime | Ngày sinh độc giả | Not null |
| 7 | gioitinh | int | Giới tính độc giả | Not null |
| 8 | sđt | string | SĐT độc giả | Not null |
| 9 | email | string | Email độc giả | Not null |

**Bảng 7. DauSach**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã ngôn ngũ | Khóa chính |
| 2 | tieude | nvarchar(30) | Tiêu đề của sách | Not null |
| 3 | mota | Nvarchar(200) | Mô tả về sách |  |
| 4 | gia | int | Giá của sách | Check(gia >=0) |
| 5 | ngayxuatban | date | Ngày xuất bản sách |  |
| 6 | hinhanh | nvarchar(100) | Hình ảnh về sách | Check(loai=0 or loai=1) |
| 7 | loai | int | Loại sách | Not null |
| 8 | trangthai | int | Trạng thái của sách | Not null |
| 9 | maNXB | int | Mã của NXB | Khóa ngoại |
| 10 | maNgonNgu | int | Mã của ngôn ngữ | Khóa ngoại |
| 11 | maTheLoai | int | Mã thể loại của sách | Khóa ngoại |

**Bảng 8. HoSo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã hồ sơ | Khóa chính |
| 2 | ten | nvarchar(20) | Tên nhân viên | Not null  Check(ten NOT LIKE '%[^a-zA-Z ]%') |
| 3 | ho | nvarchar(20) | Họ nhân viên | Not null  Check(ho NOT LIKE '%[^a-zA-Z ]%') |
| 4 | diachi | nvarchar(200) | Địa chỉ của nhân viên |  |
| 5 | soDT | nvarchar(10) | Số ĐT của nhân viên | CHECK (DATALENGTH(soDT) = 10 AND(soDT NOT LIKE '%[^0-9]%' ) |
| 6 | hinhanh | nvarchar(100) | Hình ảnh của nhân viên |  |
| 7 | email | nvarchar(100) | Email của nhân viên | Not null |
| 8 | ngaysinh | date | Ngày sinh của nhân viên |  |
| 9 | maNhanVien | int | Mã nhân viên | Khóa ngoại |

**Bảng 9. NhanVien**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã tài khoản | Khóa chính |
| 2 | tenDangNhap | nvarchar(30) | Tên tài khoản | Not null  check(tenDangNhap NOT LIKE '%[^a-zA-Z ]%') |
| 3 | matkhau | nvarchar(100) | Mật khẩu | Not null  Check(DATALENGTH(matkhau) >= 8) |
| 4 | trangthai | int | Trạng thái của tài khoản | Not null  Check (trangthai=0 or trangthai=1) |
| 5 | maHoSo | int | Mã hồ sơ | Not null  Khóa ngoại |

**Bảng 10. Muon**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | Int | Mã mượn | Khóa chính |
| 2 | ngaymuon | date | Ngày mượn sách | Not null |
| 3 | trangthai | int | Trạng thái của sách | Not null  Check(trangthai=0 or trangthai=1) |
| 4 | tienphat | int | Tiền phạt khi trả sách muộn | Check(tienphat>0) |
| 5 | maNhanVien | int | Mã nhân viên cho mươn | Not null  Khóa ngoại |
| 6 | maDocGia | int | Mã độc giả mượn | Not null  Khóa ngoại |

**Bảng 11. tacgia\_sach**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | maDauSach | int | Mã đầu sách | Khóa chính  Khóa ngoại |
| 2 | maTacGia | int | Mã tác giả | Khóa chính  Khóa ngoại |

**Bảng 12. vaitro\_nhanvien**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | maNhanVien | int | Mã nhân viên | Khóa chính  Khóa ngoại |
| 2 | maVaiTro | int | Mã vai trò | Khóa chính  Khóa ngoại |

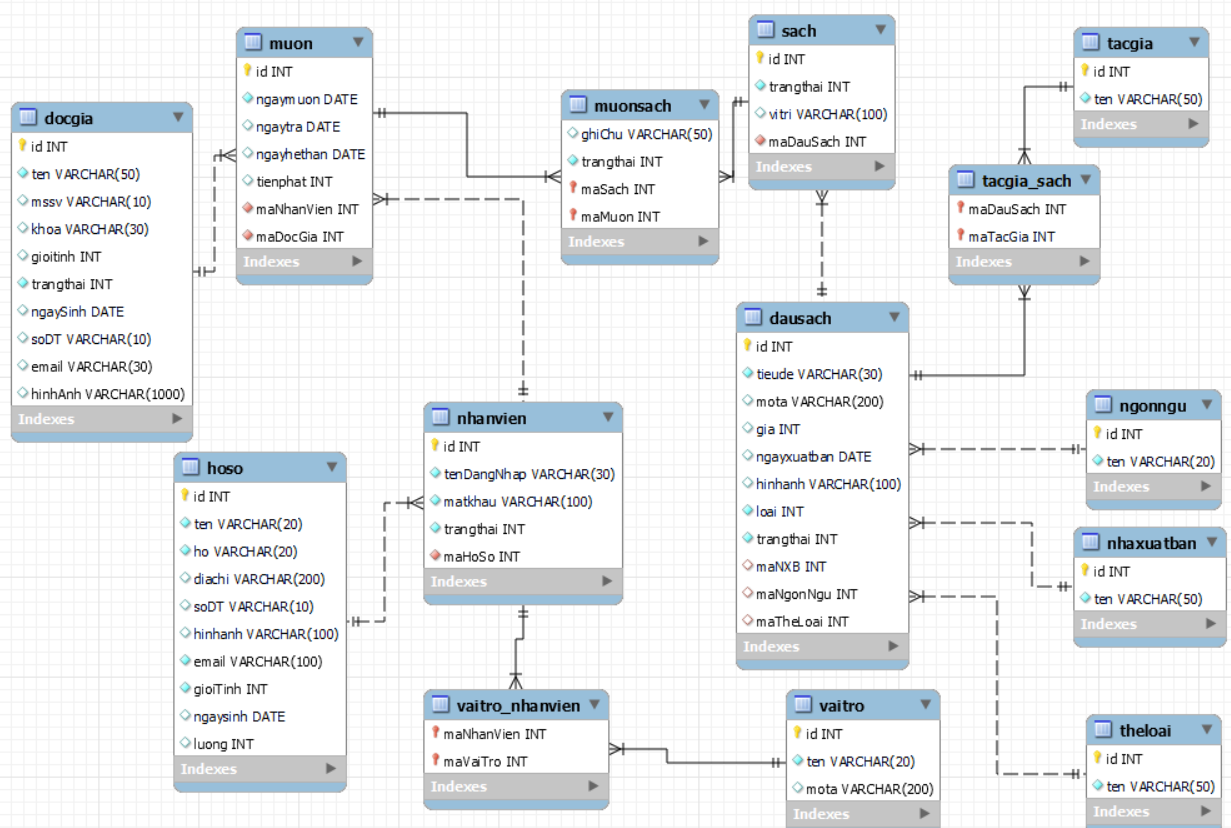
**Bảng 13. Sach**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | id | int | Mã sách | Khóa chính |
| 2 | trangthai | int | Trạng thái của sách | Not null  Check(0 or 1) |
| 3 | vitri | nvarchar(100) | Vị trí của sách |  |
| 4 | maDauSach | int | Mã đầu sách | Not null Khóa ngoại |

**Bảng 14. MuonSach**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | ngayhethan | date | Ngày hết hạn mượn sách | Not null |
| 2 | ngaytra | date | Ngày trả sách | Not null |
| 3 | maSach | int | Mã sách | Khóa chính  Khóa ngoại |
| 4 | maMuon | int | Mã mượn | Khóa chính  Khóa ngoại |

## Thiết kế mức logic



# Cài đặt cơ sở dữ liệu và ràng buộc

## Tạo bảng TheLoai:

CREATE TABLE TheLoai

(

id INT IDENTITY(1,1),

ten NVARCHAR(50) NOT NULL,

CONSTRAINT PK\_TheLoai PRIMARY KEY(id),

CONSTRAINT CHK\_TheLoai CHECK (ten NOT LIKE '%[^a-zA-Z ]%')

);

## Tạo bảng NgonNgu

CREATE TABLE NgonNgu

(

id INT IDENTITY(1,1),

ten NVARCHAR(20) NOT NULL,

CONSTRAINT PK\_NgonNgu PRIMARY KEY(id),

CONSTRAINT CHK\_NgonNgu CHECK (ten like '%[^a-zA-Z ]%')

);

## Tạo bảng TacGia

CREATE TABLE TacGia

(

id INT IDENTITY(1,1),

ten NVARCHAR(50) NOT NULL,

CONSTRAINT PK\_TacGia PRIMARY KEY(id),

CONSTRAINT CHK\_TacGia CHECK (ten NOT LIKE '%[^a-zA-Z ]%')

);

## Tạo bảng NhaXuanBan

CREATE TABLE NhaXuatBan

(

id INT IDENTITY(1,1),

ten NVARCHAR(50) NOT NULL,

CONSTRAINT PK\_NhaXuatBan PRIMARY KEY(id)

);

## Tạo bảng VaiTro

CREATE TABLE VaiTro

(

id INT IDENTITY(1,1),

ten NVARCHAR(20) NOT NULL,

mota NVARCHAR(200),

CONSTRAINT PK\_VaiTro PRIMARY KEY(id)

);

## Tạo bảng DocGia

CREATE TABLE DocGia

(

id INT IDENTITY(1,1),

ten NVARCHAR(50) NOT NULL,

mssv NVARCHAR(10),

khoa NVARCHAR(30),

trangthai INT NOT NULL,

CONSTRAINT PK\_DocGia PRIMARY KEY(id)

);

## Tạo bảng DauSach

CREATE TABLE DauSach

(

id INT IDENTITY(1,1),

tieude NVARCHAR(30) NOT NULL,

mota NVARCHAR(200),

gia INT,

ngayxuatban DATE,

hinhanh NVARCHAR(100),

loai INT NOT NULL,

trangthai INT NOT NULL,

maNXB INT,

maNgonNgu INT,

maTheLoai INT,

CONSTRAINT PK\_DauSach PRIMARY KEY(id),

CONSTRAINT FK\_NhaXB\_DauSach FOREIGN KEY (maNXB) REFERENCES NhaXuatBan(id),

CONSTRAINT FK\_NgonNgu\_DauSach FOREIGN KEY (maNgonNgu) REFERENCES NgonNgu(id),

CONSTRAINT FK\_TheLoai\_DauSach FOREIGN KEY (maTheLoai) REFERENCES TheLoai(id),

CONSTRAINT CHK\_DauSach CHECK (gia>=0 AND (loai=0 OR loai=1))

);

## Tạo bảng HoSo

CREATE TABLE HoSo

(

id INT IDENTITY(1,1),

ten NVARCHAR(20) NOT NULL,

ho NVARCHAR(20) NOT NULL,

diachi NVARCHAR(200),

soDT NVARCHAR(10),

hinhanh NVARCHAR(100),

email NVARCHAR(100) NOT NULL,

ngaysinh Date,

CONSTRAINT PK\_HoSo PRIMARY KEY(id),

CONSTRAINT CHK\_HoSo CHECK (DATALENGTH(soDT) = 10 AND(soDT NOT LIKE '%[^0-9]%'

AND ten NOT LIKE '%[^a-zA-Z ]%'

AND ho NOT LIKE '%[^a-zA-Z ]%'))

);

## Tạo bảng NhanVien

CREATE TABLE NhanVien

(

id INT IDENTITY(1,1),

tenNhanVien NVARCHAR(30) NOT NULL,

matkhau NVARCHAR(100) NOT NULL,

trangthai INT NOT NULL,

maHoSo INT NOT NULL,

CONSTRAINT PK\_NhanVien PRIMARY KEY(id),

CONSTRAINT FK\_HoSo\_NhanVien FOREIGN KEY (maHoSo) REFERENCES HoSo(id),

CONSTRAINT CHK\_NhanVien CHECK (DATALENGTH(matkhau) >= 8 AND (tenNhanVien NOT LIKE '%[^a-zA-Z ]%')

AND (trangthai = 0 OR trangthai =1))

);

## Tạo bảng Muon

CREATE TABLE Muon

(

id INT IDENTITY(1,1),

ngaymuon DATE NOT NULL,

ngaytra DATE,

trangthai INT NOT NULL,

tienphat INT,

maNhanVien INT NOT NULL,

maDocGia INT NOT NULL,

CONSTRAINT PK\_Muon PRIMARY KEY(id),

CONSTRAINT FK\_Muon\_NhanVien FOREIGN KEY (maNhanVien) REFERENCES NhanVien(id),

CONSTRAINT FK\_Muon\_DocGia FOREIGN KEY (maDocGia) REFERENCES DocGia(id),

CONSTRAINT CHK\_Muon CHECK (tienphat >= 0 AND (trangthai = 0 OR trangthai =1))

);

## Tạo bảng tacgia\_sach

CREATE TABLE tacgia\_sach

(

maDauSach INT NOT NULL,

maTacGia INT NOT NULL,

CONSTRAINT PK\_tacgia\_sach PRIMARY KEY (maDauSach, maTacGia),

CONSTRAINT FK\_DauSach FOREIGN KEY (maDauSach) REFERENCES DauSach(id),

CONSTRAINT FK\_TacGia FOREIGN KEY (maTacGia) REFERENCES TacGia(id)

);

## Tạo bảng vaitro\_nhanvien

CREATE TABLE vaitro\_nhanVien

(

maNhanVien INT NOT NULL,

maVaiTro INT NOT NULL,

CONSTRAINT PK\_VaiTroNhanVien PRIMARY KEY (maNhanVien, maVaiTro),

CONSTRAINT FK\_VaiTro FOREIGN KEY (maVaiTro) REFERENCES VaiTro(id),

CONSTRAINT FK\_NhanVien FOREIGN KEY (maNhanVien) REFERENCES NhanVien(id)

);

## Tạo bảng Sach

CREATE TABLE Sach

(

id INT IDENTITY(1,1),

trangthai INT NOT NULL,

vitri NVARCHAR(100),

maDauSach INT NOT NULL,

CONSTRAINT PK\_Sach PRIMARY KEY (id),

CONSTRAINT FK\_DauSach\_Sach FOREIGN KEY (maDauSach) REFERENCES DauSach(id),

CONSTRAINT CHK\_Sach CHECK (trangthai = 0 OR trangthai = 1)

);

## Tạo bảng MuonSach

CREATE TABLE MuonSach

(

ngayhethan DATE,

ngaytra INT NOT NULL,

maSach INT NOT NULL,

maMuon INT NOT NULL,

CONSTRAINT PK\_MuonSach PRIMARY KEY (maSach, maMuon),

CONSTRAINT FK\_Sach FOREIGN KEY (maSach) REFERENCES Sach(id),

CONSTRAINT FK\_Muon FOREIGN KEY (maMuon) REFERENCES Muon(id)

);

# Cài đặt các chức năng

## View

-- Tạo view nhanvien

CREATE VIEW NHANVIEN\_VIEW AS

SELECT HoSo.id, HoSo.ten, Hoso.email

FROM HoSo INNER JOIN NhanVien ON HoSo.id=NhanVien.maHoSo

INNER JOIN vaitro\_nhanVien ON NhanVien.id=vaitro\_nhanVien.maNhanVien

INNER JOIN VaiTro ON vaitro\_nhanVien.maVaiTro=VaiTro.id

WHERE VaiTro.ten='nhan vien'

GO

## Trigger

-- if ngayxuatban > current time => rollback

-- or ngayxuatban is older than 1900 => rollback also

CREATE TRIGGER trig\_namxuatban ON DauSach

FOR INSERT, UPDATE

AS

DECLARE @ngayxuatban DATE;

SELECT @ngayxuatban = DauSach.ngayxuatban FROM INSERTED DauSach

IF YEAR(GETDATE()) - YEAR(@ngayxuatban) < 0 OR YEAR(@ngayxuatban) < 1900

BEGIN

ROLLBACK

END

GO

-- instead of delete dausach, change status from 1 to 0 (unactive )

CREATE TRIGGER trig\_trangthai\_dausach ON DauSach

INSTEAD OF DELETE

AS

DECLARE @trangthai INT;

IF (SELECT trangthai FROM DELETED DauSachDel) != 0

BEGIN

DECLARE @id INT

SELECT @id = id FROM DELETED

UPDATE DauSach

SET trangthai = 0

WHERE DauSach.id = @id;

END

GO

-- delete muon if trangthai == 1 => rollback

CREATE TRIGGER trig\_trangthai\_muon ON Muon

FOR DELETE

AS

IF (SELECT trangthai FROM DELETED ) = 1

BEGIN

ROLLBACK

END

Go

-- delete sach if trangthai == -1 => rollback

CREATE TRIGGER trig\_trangthai\_sach ON Sach

INSTEAD OF DELETE

AS

DECLARE @trangthai INT;

IF (SELECT trangthai FROM DELETED SachDel) != -1

BEGIN

DECLARE @id INT

SELECT @id = id FROM DELETED

UPDATE Sach

SET trangthai = -1

WHERE Sach.id = @id;

END

GO

-- if role is admin but age < 18 => rollback

CREATE TRIGGER trig\_HoSo ON HoSo

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @ngaysinh DATE, @role NVARCHAR(20);

SELECT @ngaysinh = ngaysinh FROM INSERTED HoSo

SELECT @role = VaiTro.ten FROM INSERTED HoSo

INNER JOIN NhanVien ON HoSo.id=NhanVien.maHoSo

INNER JOIN vaitro\_nhanVien ON NhanVien.id=vaitro\_nhanVien.maNhanVien INNER JOIN VaiTro ON vaitro\_nhanVien.maVaiTro=VaiTro.id

IF ((YEAR(GETDATE())-YEAR(@ngaysinh)<18) AND @role='admin')

BEGIN

ROLLBACK

END

END

GO

-- if trangthai == 1 => rollback

CREATE TRIGGER trigg\_enable\_nhanVien ON NhanVien

FOR DELETE

AS

BEGIN

DECLARE @trangthai INT

SELECT @trangthai = trangthai FROM NhanVien

IF (@trangthai = 1)

BEGIN

ROLLBACK

END

END

## Stored procedure

-- procedure Xem thông tin độc giả

CREATE PROCEDURE usp\_Xem\_Thong\_Tin\_Doc\_gia

AS

BEGIN

SELECT \* FROM DocGia;

END;

GO

-- procedure Thêm thông tin độc giả

CREATE PROCEDURE usp\_Them\_Doc\_Gia

@trangthai INT,

@ten NVARCHAR(50),

@mssv NVARCHAR(10),

@khoa NVARCHAR(30)

AS

BEGIN

INSERT INTO DocGia (ten,mssv,khoa,trangthai)

VALUES (@ten,@mssv,@khoa,@trangthai);

END;

GO

-- procedure Chuyển trạng thái độc giả

CREATE PROCEDURE usp\_Chuyen\_Trang\_Thai\_Doc\_gia

@id INT,

@trangthai INT

AS

BEGIN

UPDATE Docgia

SET trangthai = @trangthai

WHERE id=@id;

END;

GO

-- procedure sửa thông tin độc giả

CREATE PROC usp\_Sua\_Thong\_Tin\_Doc\_Gia

@id INT,

@trangthai INT,

@ten NVARCHAR(50),

@mssv NVARCHAR(10),

@khoa NVARCHAR(30)

AS

BEGIN

UPDATE DocGia SET ten=@TEN, trangthai = @trangthai, mssv = @mssv, khoa = @khoa

WHERE id=@id

END;

GO

-- procedure Xem sách theo ngôn ngữ

CREATE PROCEDURE usp\_Xem\_Sach\_Theo\_Ngon\_Ngu

@id INT

AS

BEGIN

SELECT DauSach.tieude, DauSach.mota, DauSach.hinhanh FROM

DauSach INNER JOIN NgonNgu ON DauSach.maNgonNgu = NgonNgu.id

WHERE NgonNgu.id=@id

END;

GO

-- procedure Sửa thông tin nhân viên

CREATE PROC usp\_Sua\_Thong\_Tin\_Nhan\_Vien

@ID INT,

@TEN NVARCHAR(20),

@HO NVARCHAR(20),

@DIACHI NVARCHAR(20),

@SODT NVARCHAR(10),

@HINHANH NVARCHAR(100),

@EMAIL NVARCHAR(100),

@NGAYSINH DATE,

@LUONG INT

AS

BEGIN

UPDATE HoSo SET ten=@TEN, ho=@HO, diachi=@DIACHI, soDT=@SODT, hinhanh=@HINHANH, email=@EMAIL, ngaysinh=@NGAYSINH, luong = @LUONG

WHERE id=@ID

END;

GO

-- procedure Thêm tài khoản nhân viên

CREATE PROC usp\_Them\_Tai\_Khoan\_Nhan\_Vien

@TENTK NVARCHAR(30),

@MK NVARCHAR(100),

@TRANGTHAI INT,

@MAHS INT

AS

BEGIN

INSERT INTO NhanVien VALUES(@TENTK, @MK, @TRANGTHAI, @MAHS)

END;

GO

-- procedure Xem thông tin nhân viên

CREATE PROC usp\_Xem\_Thong\_Tin\_Nhan\_Vien

@ID INT

AS

BEGIN

SELECT \* FROM HoSo WHERE id=@ID

END;

GO

-- procedure chuyển trạng thái tài khoản nhân viên

CREATE PROC usp\_Chuyen\_Trang\_Thai\_Nhan\_Vien

@ID INT,

@TRANGTHAI INT

AS

BEGIN

UPDATE NhanVien SET trangthai=@TRANGTHAI WHERE id=@ID

END;

GO

-- procedure Xem thông tin đầu sách

CREATE PROCEDURE usp\_Xem\_Dau\_Sach

AS

BEGIN

SELECT \* FROM DauSach;

END;

GO

-- procedure Thêm thông tin đầu sách

CREATE PROCEDURE usp\_Them\_Dau\_Sach

@tieude NVARCHAR(30),

@mota NVARCHAR(200),

@gia INT,

@ngayxuatban DATE,

@hinhanh NVARCHAR(100),

@loai INT,

@trangthai INT,

@maNXB INT,

@maNgonNgu INT,

@maTheLoai INT

AS

BEGIN

INSERT INTO DauSach (

tieude,

mota,

gia,

ngayxuatban,

hinhanh,

loai,

trangthai,

maNXB,

maNgonNgu,

maTheLoai

)

VALUES (

@tieude,

@mota,

@gia,

@ngayxuatban,

@hinhanh,

@loai,

@trangthai,

@maNXB,

@maNgonNgu,

@maTheLoai

);

END;

GO

-- procedure Chuyển trạng thái đầu sách ( Cho mượn, không cho mượn)

CREATE PROCEDURE usp\_Chuyen\_Trang\_Thai\_Dau\_Sach

@id INT,

@trangthai INT

AS

BEGIN

UPDATE DauSach

SET trangthai = @trangthai

WHERE id=@id;

END;

GO

-- procedure Liệt kê sách theo trạng thái

CREATE PROCEDURE usp\_Liet\_Ke\_Sach\_Theo\_Trang\_Thai

@trangthai INT

AS

BEGIN

SELECT \* FROM DauSach WHERE trangthai=@trangthai;

END;

GO

-- procedure Liệt kê những độc giả đang mượn sách

CREATE PROCEDURE usp\_Liet\_Ke\_Doc\_Gia\_Dang\_Muon\_Sach

AS

BEGIN

SELECT DocGia.ten, DocGia.mssv, DocGia.khoa FROM

DocGia INNER JOIN Muon ON DocGia.id = Muon.maDocGia;

END;

GO

-- procedure Liệt kê những độc giả đang mượn sách quá hạn

CREATE PROCEDURE usp\_Liet\_Ke\_Doc\_Gia\_Dang\_Muon\_Sach\_Qua\_Han

AS

BEGIN

SELECT DocGia.ten, DocGia.mssv, DocGia.khoa FROM

DocGia INNER JOIN Muon ON DocGia.id = Muon.maDocGia

WHERE MONTH(GETDATE())-MONTH(Muon.ngaymuon) >=6;

END;

GO

-- procedure Xem sách theo tác giả

CREATE PROCEDURE usp\_Xem\_Sach\_Theo\_Tac\_Gia

@id INT

AS

BEGIN

SELECT DauSach.tieude, DauSach.mota, DauSach.hinhanh FROM

TacGia INNER JOIN tacgia\_sach ON TacGia.id = tacgia\_sach.maTacGia

INNER JOIN DauSach ON DauSach.id = tacgia\_sach.maDauSach

WHERE TacGia.id = @id;

END;

GO

-- procedure Xem sách theo danh mục

CREATE PROCEDURE usp\_Xem\_Sach\_Theo\_Danh\_Muc

@id INT

AS

BEGIN

SELECT DauSach.tieude, DauSach.mota, DauSach.hinhanh FROM

DauSach INNER JOIN TheLoai ON DauSach.maTheLoai = TheLoai.id

WHERE TheLoai.id = @id;

END;

GO

-- procedure xem thông tin phiếu mượn

CREATE PROCEDURE usp\_Xem\_Thong\_Tin\_Phieu\_Muon

@id INT

AS

BEGIN

SELECT \* FROM Muon

WHERE id = @id;

END;

GO

-- procedure thêm thông tin phiếu mượn

CREATE proc usp\_Them\_Thong\_Tin\_Phieu\_Muon

@ngaymuon DATE,

@ngaytra DATE,

@trangthai INT,

@tienphat INT,

@maNhanVien INT,

@maDocGia INT

AS

BEGIN

INSERT INTO Muon VALUES(@ngaymuon, @ngaytra, @trangthai, @tienphat,@maNhanVien,@maDocGia)

END;

GO

-- procedure xóa thông tin phiếu mượn quá hạn ( trên 1 năm )

CREATE PROC usp\_Xoa\_Phieu\_Muon\_Qua\_Han

AS

BEGIN

DELETE FROM Muon WHERE DATEDIFF(day, GETDATE(), Muon.ngaytra) > 365

END;

GO

-- procedure đổi mật khẩu nhân viên

CREATE PROC usp\_Doi\_Mat\_Khau\_Nhan\_Vien

@ID INT,

@MATKHAU NVARCHAR(100)

AS

BEGIN

UPDATE NhanVien SET matkhau=@MATKHAU WHERE id=@ID

END;

GO

-- procedure xem đầu sách theo nhà xuất bản

CREATE PROCEDURE usp\_Xem\_Dau\_Sach\_Theo\_Nha\_Xuat\_Ban

@maNXB INT

AS

BEGIN

SELECT \* FROM DauSach INNER JOIN NhaXuatBan ON DauSach.maNXB = NhaXuatBan.id

WHERE NhaXuatBan.id = @maNXB;

END;

GO

## Function

-- function trả về tổng lương nhân viên

CREATE FUNCTION fn\_Tong\_Luong\_Nhan\_Vien()

RETURNS INT AS

BEGIN

DECLARE @salary INT;

SELECT @salary = COUNT(HoSo.luong) FROM NhanVien

INNER JOIN HoSo ON NhanVien.maHoSo = HoSo.id;

RETURN @salary;

END;

GO

-- function trả về tổng số lượng sách theo trạng thái

CREATE FUNCTION fn\_Tong\_Sach\_Theo\_Trang\_Thai(

@ID\_DAU\_SACH INT,

@TRANG\_THAI INT

)

RETURNS INT

AS

BEGIN

DECLARE @NUMBER INT;

SELECT @NUMBER=COUNT(\*) FROM Sach WHERE maDauSach=@ID\_DAU\_SACH and trangthai=@TRANG\_THAI;

RETURN @NUMBER

END;

GO

-- function trả về vai trò theo nhân viên

CREATE FUNCTION fn\_Vai\_Tro\_Nhan\_Vien()

RETURNS NVARCHAR AS

BEGIN

DECLARE @vaitro NVARCHAR;

SELECT @vaitro = VaiTro.ten FROM NhanVien

INNER JOIN vaitro\_nhanVien ON NhanVien.id = vaitro\_nhanVien.maNhanVien

INNER JOIN VaiTro ON vaitro\_nhanVien.maVaiTro=VaiTro.id;

RETURN @vaitro;

END;

GO

-- function kiểm tra đăng nhập

CREATE PROC usp\_Kiem\_Tra\_Dang\_Nhap

@tenDangNhap NVARCHAR(40),

@matKhau Nvarchar(100)

AS

BEGIN

DECLARE @maNhanVien INT

SELECT @maNhanVien = NhanVien.id FROM NhanVien WHERE NhanVien.tenDangNhap = @tenDangNhap AND NhanVien.matkhau = @matKhau

IF (@maNhanVien IS NOT NULL)

SELECT @maNhanVien

ELSE

SELECT -1

END