

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM
KHOA: HỆ THỐNG THÔNG TIN VÀ VIỆN THĂM



ĐỒ ÁN MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
ĐỀ TÀI: ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ
THỐNG QUẢN LÝ THƯ VIỆN ONLINE CỦA TRƯỜNG ĐẠI HỌC TÀI
NGUYÊN VÀ MÔI TRƯỜNG

Giảng viên hướng dẫn: **ThS. Phạm Trọng Huỳnh**

Sinh viên thực hiện: **Nhóm 1:**

Phạm Minh Tuấn

Hà Chí Bảo

Nguyễn Mạnh Hùng Vĩ

Lớp: **09_CNPM_03**

Tp. Hồ Chí Minh, tháng 03 năm 2023

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM
KHOA: HỆ THỐNG THÔNG TIN VÀ VIỆN THĂM



ĐỒ ÁN MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
ĐỀ TÀI: ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ
THỐNG QUẢN LÝ THƯ VIỆN ONLINE CỦA TRƯỜNG ĐẠI HỌC TÀI
NGUYÊN VÀ MÔI TRƯỜNG

Giảng viên hướng dẫn: **ThS. Phạm Trọng Huỳnh**

Sinh viên thực hiện: **Nhóm 1:**

Phạm Minh Tuấn

Hà Chí Bảo

Nguyễn Mạnh Hùng Vĩ

Lớp: **09_CNPM_03**

Tp. Hồ Chí Minh, tháng 03 năm 2023

Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày....tháng.... năm 2023

Giảng viên

Phạm Trọng Huỳnh

MỤC LỤC	
HÌNH ẢNH.....	3
LỜI NÓI ĐẦU	4
CHƯƠNG 1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	5
1.1. Lời giới thiệu:	5
1.2. Phân tích hệ thống quản lý thư viện online của trường đại học tài nguyên và môi trường:.....	5
1.2.1. Thực trạng của hệ thống quản lý thư viện online của trường đại học tài nguyên và môi trường.	5
1.2.2. Biểu đồ phân cấp chức năng:	5
1.2.3. Biểu đồ luồng dữ liệu:	6
1.2.4. Phân loại các thuộc tính vào một tập thực thể.....	11
CHƯƠNG 2. ỨNG DỤNG LÝ THUYẾT	15
2.1. Xây dựng CSDL trên HQT CSDL:	15
2.1.1. Ghi câu lệnh tạo CSDL, tạo bảng, tạo khóa:	15
2.1.2. Cấu trúc bảng:	19
2.1.3. Dữ liệu của mỗi bảng:	21
2.1.4. Màn hình Diagram liên kết các bảng:.....	26
2.2. Thao tác trên CSDL trên HQT CSDL:	26
2.2.1. Tạo và xóa tài khoản người dùng đăng nhập quyền sql server + Bảng công cụ:	26
2.2.2. Viết câu lệnh Insert:	28
2.2.3. Viết câu lệnh update:	29
2.3. Store Procedure, Funtion, Trigger:	30
2.3.1. Store Procedure:	30
2.3.2. Function:	34
2.3.3. Trigger:	38
2.4. Bảo mật, phân quyền:	40
Chương 3: KẾT LUẬN	44
3.1. Những kết quả đạt được của đồ án:	44
3.2. Nhược điểm của đồ án:	44
PHỤ LỤC	45
TÀI LIỆU THAM KHẢO.....	49

HÌNH ẢNH

Hình 1. Bảng biểu đồ phân cấp chức năng	6
Hình 2. Biểu đồ luồng dữ liệu ngữ cảnh.....	6
Hình 3. Biểu đồ luồng dữ liệu mức 0	7
Hình 4. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý độc giả	7
Hình 5. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý tài liệu	8
Hình 6. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý mượn trả	8
Hình 7. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng báo cáo thống kê.....	9
Hình 8. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý độc giả	9
Hình 9. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý tài liệu	10
Hình 10. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý mượn trả	10
Hình 11. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng báo cáo thống kê	11
Hình 12. Bảng phân quyền	41

LỜI NÓI ĐẦU

Quản lý thư viện sách là một công việc phức tạp, đòi hỏi người quản lý cần phải có sự tỉ mỉ, chu đáo và một phương pháp quản lý khoa học. Tuy nhiên, khi quy mô thư viện tăng lên, số lượng sách nhiều thêm và bạn đọc cũng có thêm nhiều nhu cầu khác nhau, công việc của người quản lý lại càng gặp thêm nhiều vấn đề.

Với thực trạng hiện nay đó là công việc quản lý diễn ra trên giấy tờ, nhưng khối lượng dữ liệu thì ngày một lớn. Vì vậy các hệ thống quản lý ra đời nhằm giải quyết những khó khăn trên giúp cho công việc quản lý trở nên đơn giản, hiệu quả và chính xác hơn.

Xuất phát từ nhu cầu đó, nhóm chúng em thực hiện đề tài “Xây dựng hệ thống quản lý thư viện online của trường đại học Tài nguyên và Môi trường”. Hệ thống này sẽ giúp: Cập nhật thông tin, số lượng cụ thể của mỗi đầu mục sách, mỗi cuốn sách và mỗi hạng mục sách. Bạn có thể dễ dàng hơn trong việc tìm kiếm và cập nhật thông tin của từng cuốn sách. Cập nhật thông tin của từng độc giả. Cập nhật tình trạng sử dụng, mượn, trả sách và thông tin kỳ hạn thẻ thư viện của từng độc giả. Cập nhật vị trí, số lượng của từng loại sách, hỗ trợ người quản lý có thể dễ dàng sắp xếp chúng vào những vị trí nhất định. Công việc sắp xếp và quản lý sách có thể được thực hiện một cách nhanh chóng và khoa học hơn. Bạn có thể đồng bộ hóa thông tin dữ liệu trong thư viện để có thể tìm kiếm, cập nhật thông tin từng đầu sách nhanh hơn. Bạn có thể thao tác nhanh trong việc thực hiện những yêu cầu của từng độc giả.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Phạm Trọng Huỳnh đã giúp đỡ và hướng dẫn tận tình chúng em để hoàn thành đề tài “Xây dựng hệ thống quản lý thư viện online của trường đại học Tài nguyên và Môi trường”, cũng như Thư viện của nhà trường đã tạo điều kiện cho chúng em hoàn thành bài báo cáo.

Do kinh nghiệm và kiến thức chuyên môn còn hạn chế cho nên không tránh khỏi thiếu sót. Chúng em rất mong nhận được sự góp ý từ các thầy cô hướng dẫn và bạn đọc để chúng mình có thể rút thêm kinh nghiệm hoàn thành các bài báo cáo về sau.

CHƯƠNG 1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1.1. Lời giới thiệu:

Thư viện nơi mà có rất nhiều sinh viên đến để mượn sách và điều này kéo theo việc quản lý số lượng thông tin ngày càng cao. Việc quản lý sinh viên mượn sách theo cách thủ công gặp một số hạn chế sau:

- Quản lý bằng sổ sách, thủ thư cần phải ghi chép rất nhiều dẫn đến tình trạng nhầm lẫn và khó kiểm soát.
- Quản lý sách với số lượng lớn nếu không chú ý sẽ gây ra sai sót và nhầm lẫn.
- Rất khó khăn và tốn rất nhiều thời gian trong việc tìm kiếm sách.

Quản lý thư viện là một chuỗi công việc rất vất vả và tốn nhiều công sức. Việc tin học hoá trong bài toán quản lý thư viện sẽ giúp việc quản lý trở nên đơn giản và đặc biệt là tính chính xác cao. Đặc biệt tin học hoá trong bài toán quản lý sẽ giúp việc truy vấn thông tin được nhanh chóng theo yêu cầu khác nhau.

Và sự ra đời của ứng dụng này phục vụ cho công tác nghiệp vụ của con người làm giảm thiểu tối đa những vất vả trong công việc.

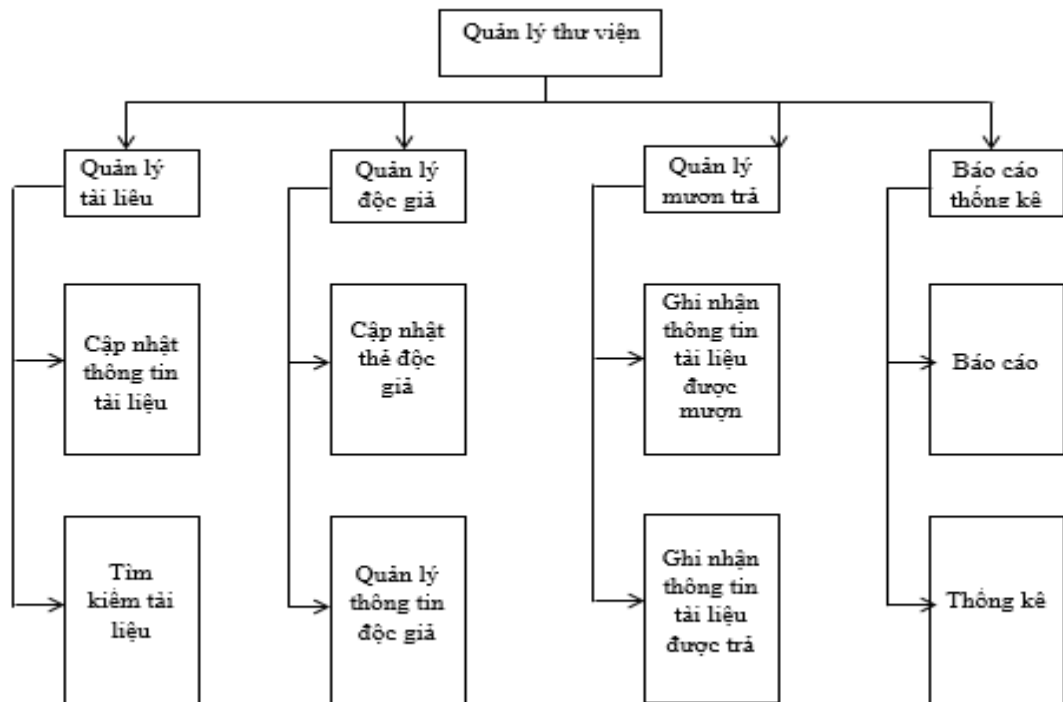
Nhận thấy việc xây dựng chương trình quản lý sách thư viện có thể giúp khắc phục khó khăn trên đồng thời được sự hướng dẫn của thầy. Bởi vậy, em quyết định tìm hiểu đề tài “**ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ THỐNG QUẢN LÝ THƯ VIỆN ONLINE CỦA TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**”.

1.2. Phân tích hệ thống quản lý thư viện online của trường đại học tài nguyên và môi trường:

1.2.1. Thực trạng của hệ thống quản lý thư viện online của trường đại học tài nguyên và môi trường.

Hệ thống quản lý thư viện giúp quản lý chặt chẽ các thông tin về các sinh viên trong việc mượn trả sách. Các thông tin này có mối liên hệ, liên kết với nhau từ đó giúp sinh viên dễ dàng tìm thấy tài liệu tương thích nhất

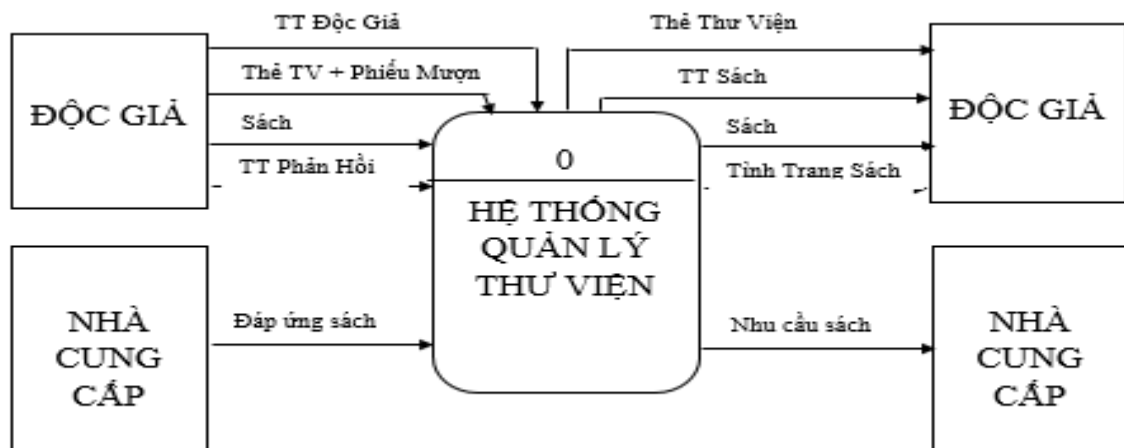
1.2.2. Biểu đồ phân cấp chức năng:



Hình 1. Bảng biểu đồ phân cấp chức năng

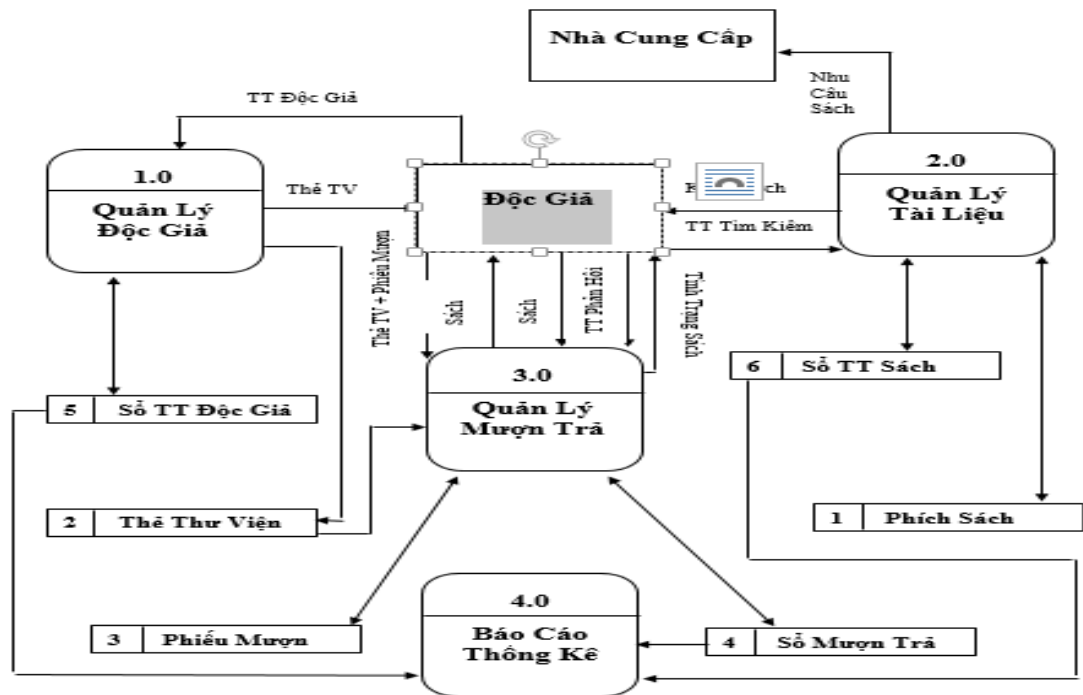
1.2.3. Biểu đồ luồng dữ liệu:

- Biểu đồ luồng dữ liệu ngữ cảnh:



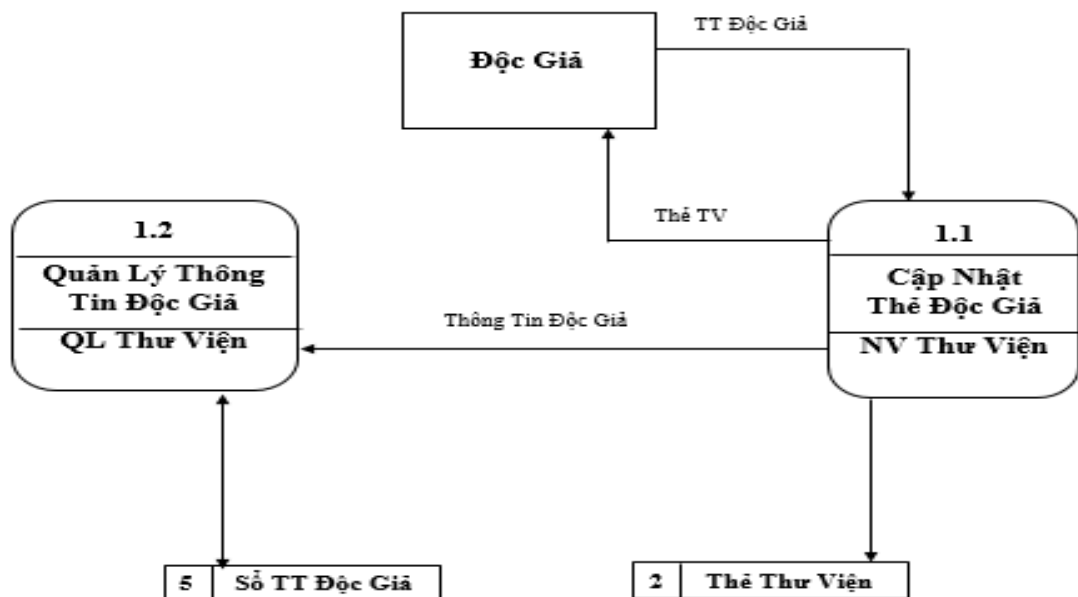
Hình 2. Biểu đồ luồng dữ liệu ngữ cảnh

- DFD mức 0



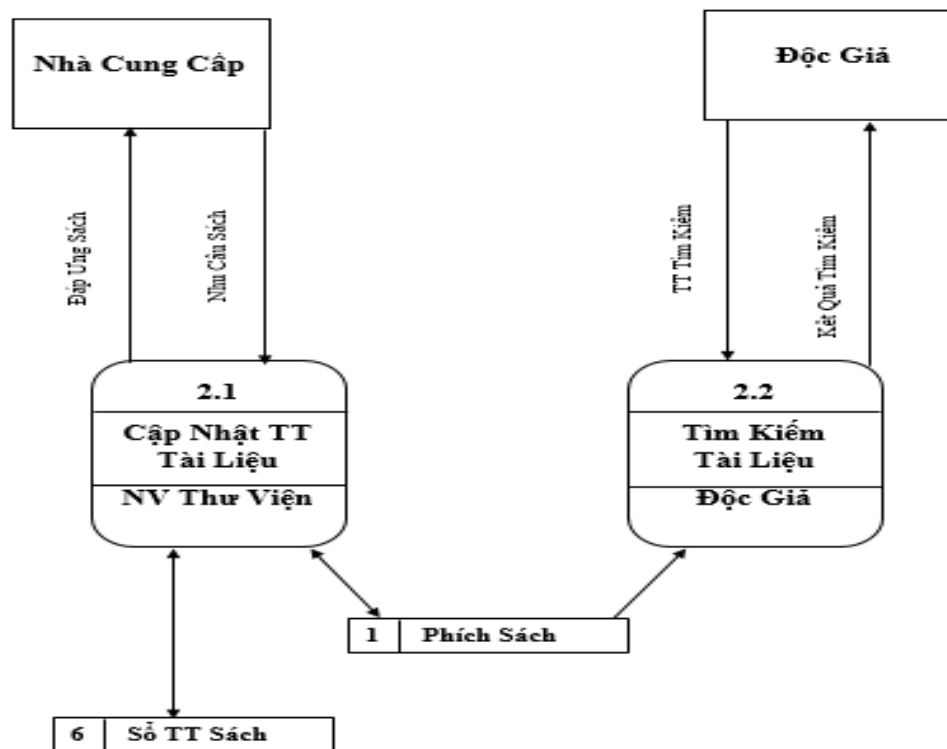
Hình 3. Biểu đồ luồng dữ liệu mức 0

- Biểu đồ luồng dữ liệu vật lý mức 1:
 - Chức năng quản lý độc giả:



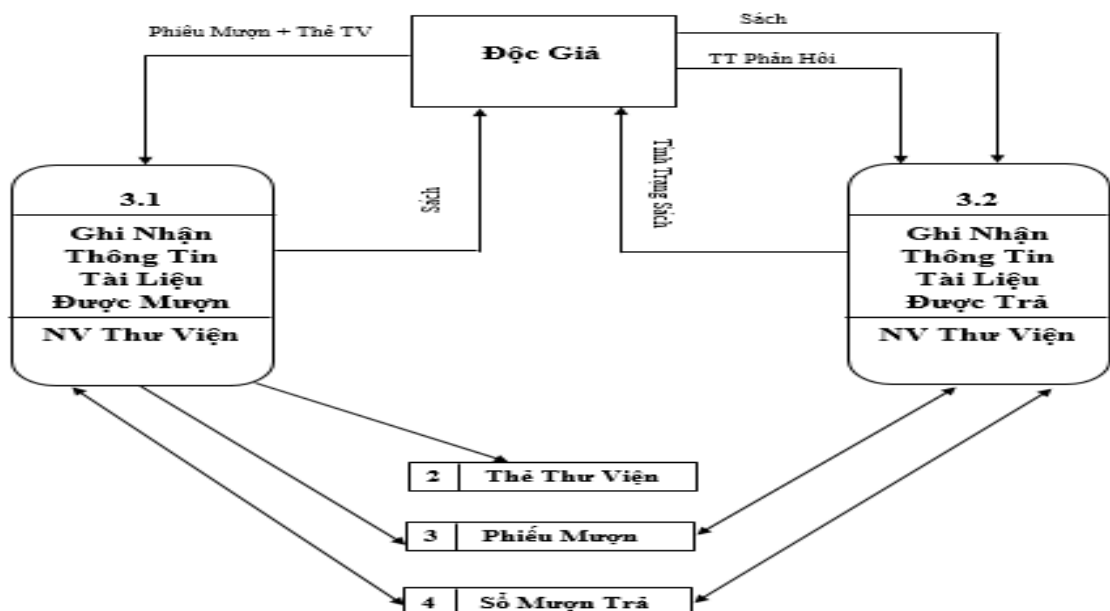
Hình 4. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý độc giả

- Chức năng quản lý tài liệu:



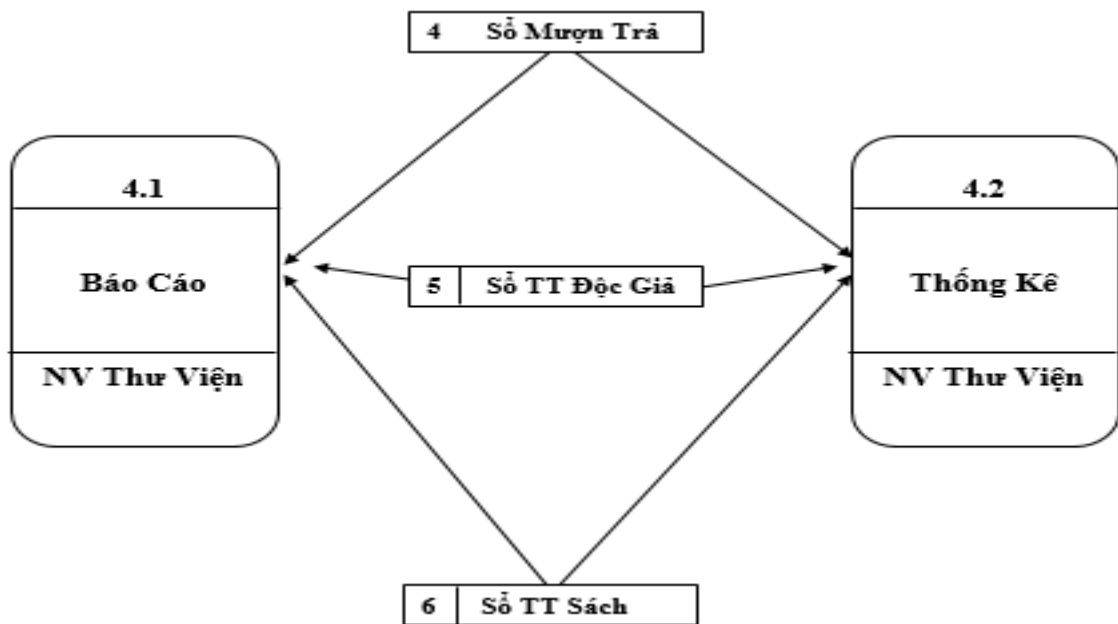
Hình 5. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý tài liệu

- Chức năng quản lý trả mượn:



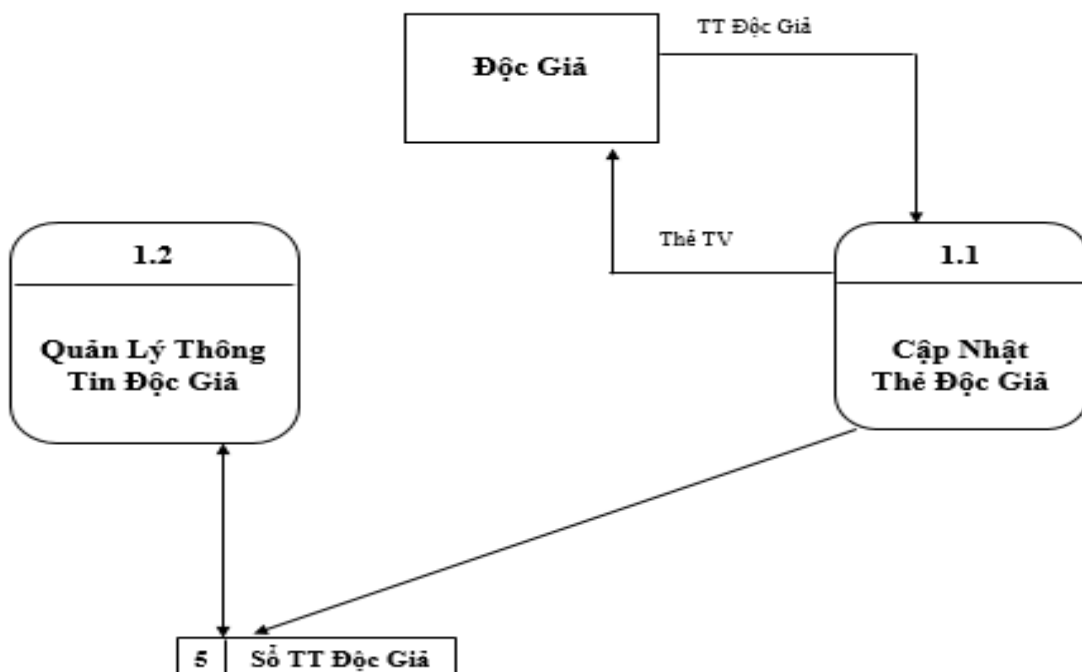
Hình 6. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng quản lý mượn trả

- Chức năng báo cáo thống kê:



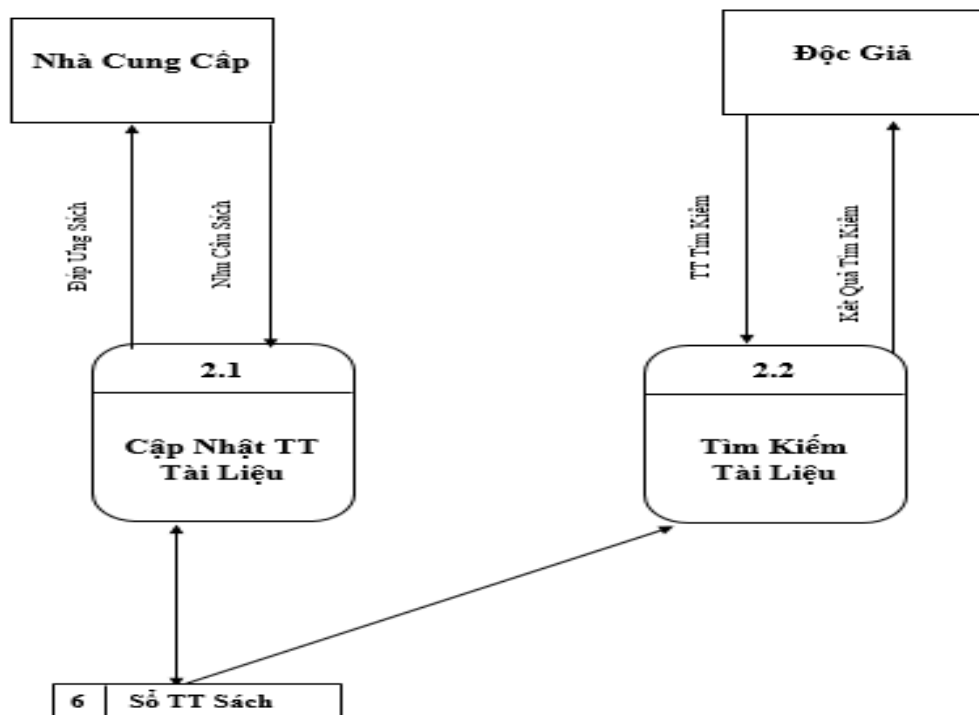
Hình 7. Biểu đồ luồng dữ liệu vật lý mức 1 của chức năng báo cáo thống kê

- Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic:
 - Chức năng quản lý độc giả:



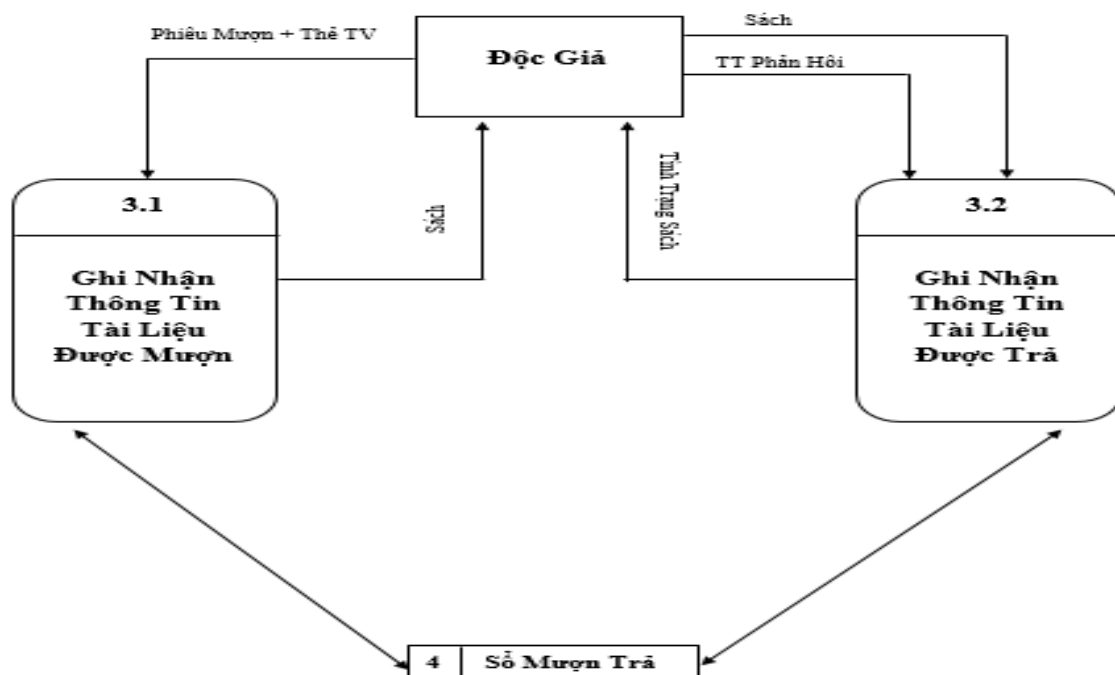
Hình 8. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý độc giả

- Chức năng quản lý tài liệu:



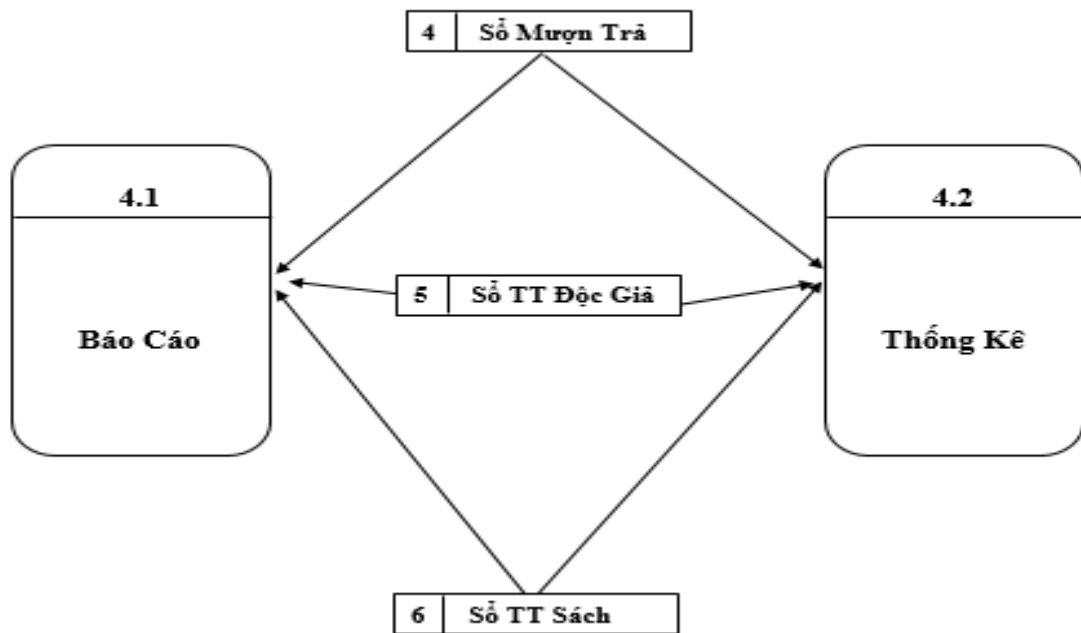
Hình 9. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý tài liệu

- Chức năng quản lý mượn trả:



Hình 10. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng quản lý mượn trả

- Chức năng báo cáo thống kê:



Hình 11. Biểu đồ luồng dữ liệu vật lý sang biểu đồ logic của chức năng báo cáo thống kê

1.2.4. Phân loại các thuộc tính vào một tập thực thể

a) Bảng nhân viên:

Nhân Viên				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaNV	Int	Not null	Mã nhân viên	Identity(1,1)
TenNV	Nvarchar(80)	Not null	Tên nhân viên	
ChucVu	Bit	Not null	Chức vụ (0 là nhân viên, 1 là trưởng kho)	Default 0
TaiKhoan	Varchar(50)	Not null	Tài khoản đăng nhập phần mềm	

MatKhou	Varchar(50)	Not null	Mật khẩu đăng nhập phần mềm	
---------	-------------	----------	-----------------------------	--

b) Bảng chủ đề:

ChuDe				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaCD	Int	Not null	Mã chủ đề	Identity(1,1)
TenCD	Nvarchar(80)	Not null	Tên chủ đề	

c) Bảng nhà xuất bản:

NhaXuatBan				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaNXB	Int	Not null	Mã nhà xuất bản	Identity(1,1)
TenNXB	Nvarchar(80)	Not null	Tên nhà xuất bản	Unique
SDT	Varchar(15)	Not null	Số điện thoại	
DiaChi	Nvarchar(100)	Not null	Địa chỉ	

d) Bảng tác giả:

TacGia				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaTG	Int	Not null	Mã tác giả	Identity(1,1)
TenTG	Nvarchar(80)	Not null	Tên tác giả	
GioiTinh	Bit	Null	Giới tính	Default 0
NgaySinh	Date	Null	Ngày sinh	

e) Bảng sách:

Sach				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaSach	Int	Not null	Mã sách	Identity(1,1)
TieuDe	Nvarchar(80)	Not null	Tiêu đề sách	Unique
MaNXB	Int	Not null	Mã nhà xuất bản	
NamXB	Date	Not null	Năm xuất bản	
MaCD	Int	Not null	Mã chủ đề	
NgonNgu	Nvarchar(50)	Not null	Ngôn ngữ	
SoTrang	Int	Not null	Số trang	
SoLuong	Int	Not null	Số lượng	
GiaTien	Bigint	Not null	Giá tiền	
TacGia	Nvarchar(150)	Not null	Tên tác giả	

f) Bảng đọc giả:

DocGia				
Tên cột	Kiểu dữ liệu		Mô tả	Default
TheDG	Varchar(15)	Not null	Thẻ đọc giả	Identity(1,1), Unique
TenDG	Nvarchar(70)	Not null	Tên đọc giả	

g) Bảng phiếu mượn:

PhieuMuon				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaPM	Int	Not null	Mã phiếu mượn	
TheDG	Varchar(15)	Not null	Thẻ đọc giả	FK(DocGia)
NgayMuon	Date	Null	Ngày mượn sách	Getdate
NgayHen	Date	Null	Ngày hẹn trả	

MaNV	Varchar(10)	Not null	Mã nhân viên	FK(NhanVien)
NgayTra	Date	Null	Ngày trả sách thực tế.	

h) Bảng phiếu mượn chi tiết:

PMCT				
Tên cột	Kiểu dữ liệu		Mô tả	Default
MaPM	Int	Not null	Mã phiếu mượn	FK(PhieuMuon)
MaSach	Int	Not null	Mã sách	FK(Sach)
TrangThai	Bit	Null	Trạng thái	Default 0 (1: đã trả sách, 0: chưa trả sách)

CHƯƠNG 2. ỨNG DỤNG LÝ THUYẾT

2.1. Xây dựng CSDL trên HQT CSDL:

2.1.1. Ghi câu lệnh tạo CSDL, tạo bảng, tạo khóa:

a) Tạo CSDL:

- Mô tả: Lệnh **CREATE DATABASE** dùng để tạo CSDL mới trong Microsoft SQL Server.

- Cú pháp: **CREATE DATABASE** Tên_CSDL

VD: [Create Database QLTV](#)

b) Tạo bảng:

- Mô tả:

- Lệnh **CREATE TABLE** dùng để tạo cấu trúc bảng trong CSDL.
- Bảng có cấu trúc gồm các cột (column) và các dòng (row).
- Trong CSDL bảng thường có khoá chính (primary key).
- Bảng dùng để lưu trữ dữ liệu, các thông tin của một đối tượng trong thực tế.
- Các bảng thường liên hệ với nhau bằng các mối quan hệ.
- Bảng được tạo trong các Schema (mặc định là schema dbo)

- Cú pháp:

```
CREATE TABLE Tên_bảng
(
    Tên_cột_1 Kiểu_dữ_liệu,
    Tên_cột_2 Kiểu_dữ_liệu,
    ...
    Tên_cột_n Kiểu_dữ_liệu
)
```

c) Tạo khóa:

- * Khóa chính:

Trong SQL Server (Transact-SQL), khóa chính là một trường hoặc kết hợp nhiều trường và được xác định là một bản ghi duy nhất. Không trường nào trong khóa chính được chứa giá trị NULL. Một bảng chỉ có duy nhất một khóa

chính. Khóa chính có thể được định nghĩa bằng lệnh **CREATE TABLE** hoặc lệnh **ALTER TABLE** .

Cú pháp tạo khóa chính bằng lệnh **CREATE TABLE**:

```
CREATE TABLE ten_bang
(
cot1 kieu_du_lieu [ NULL | NOT NULL ] [ PRIMARY KEY ],
cot2 kieu_du_lieu [ NULL | NOT NULL ],
...
);
```

Hoặc

```
CREATE TABLE ten_bang
(
cot1 kieu_du_lieu [ NULL | NOT NULL ],
cot2 kieu_du_lieu [ NULL | NOT NULL ],
...
CONSTRAINT ten_rang_buoc PRIMARY KEY (cot1, cot2, ... cot_n)
);
```

Cú pháp tạo khóa chính bằng lệnh **ALTER TABLE**:

```
ALTER TABLE ten_bang
ADD CONSTRAINT ten_rang_buoc PRIMARY KEY (cot1, cot2, ...
cot_n);
```

* Khóa ngoại:

Khóa ngoại được dùng để tăng tính tham chiếu trong cơ sở dữ liệu SQL Server. Khóa ngoại nghĩa là giá trị trong bảng này phải xuất hiện trong bảng khác.

Bảng tham chiếu gọi là bảng mẹ, còn bảng chứa khóa ngoại gọi là bảng con. Khóa ngoại trong bảng con thường tham chiếu tới khóa chính **PRIMARY KEY** trong bảng mẹ.

Khóa ngoại có thể được tạo bằng lệnh **CREATE TABLE** hoặc lệnh **ALTER TABLE**.

Tạo khóa ngoại bằng lệnh **CREATE TABLE**:

```
CREATE TABLE bang_con
(
  cot1 kieudulieu [ NULL | NOT NULL ],
  cot2 kieudulieu [ NULL | NOT NULL ],
  ...

  CONSTRAINT fk_ten
    FOREIGN KEY (cot_con1, cot_con2, ... cot_con_n)
    REFERENCES bang_me (cot_me1, cot_me2, ... cot_me_n)
    [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT
    } ]
    [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT
    } ]
);
```

Tạo khóa ngoại bằng lệnh **ALTER TABLE**:

```
ALTER TABLE bang_con
ADD CONSTRAINT fk_ten
FOREIGN KEY (cot_con1, cot_con2, ... cot_con_n)
REFERENCES bang_me (cot_me1, cot_me2, ... cot_me_n);
```

VD:

- Bảng Nhân viên:

```
CREATE TABLE NhanVien(
MaNV VARCHAR(10) PRIMARY KEY,
TenNV NVARCHAR(80) NOT NULL,
ChucVu BIT DEFAULT 0,
Email VARCHAR(60) NOT NULL,
TaiKhoan VARCHAR(50) NOT NULL UNIQUE,
MatKhau VARCHAR(32) NOT NULL,
)
```

- Bảng chủ đề

```
CREATE TABLE ChuDe(
MaCD INT IDENTITY(1,1) PRIMARY KEY,
TenCD NVARCHAR(80) UNIQUE NOT NULL,
)
```

- Bảng Nhà xuất bản

```
CREATE TABLE NhaXuatBan(
MaNXB INT IDENTITY(1,1) PRIMARY KEY,
TenNXB NVARCHAR(80) UNIQUE NOT NULL,
SDT VARCHAR(15) NOT NULL,
DiaChi NVARCHAR(100) NOT NULL
)
```

- Bảng tác giả

```
CREATE TABLE TacGia(
MaTG INT IDENTITY(1,1) PRIMARY KEY,
TenTG NVARCHAR(80) NOT NULL,
GioiTinh BIT DEFAULT 0,
NgaySinh DATE NOT NULL,
)
```

- Bảng sách

```
CREATE TABLE Sach(
MaSach INT IDENTITY(1,1) PRIMARY KEY,
TieuDe NVARCHAR(80) NOT NULL,
TacGia VARCHAR(80) NOT NULL ,
MaNXB INT NOT NULL FOREIGN KEY REFERENCES NhaXuatBan(MaNXB),
NamXB INT NOT NULL,
MaCD INT NOT NULL FOREIGN KEY REFERENCES ChuDe(MaCD),
NgonNgu NVARCHAR(40) NOT NULL,
SoTrang INT NOT NULL,
SoLuong INT NOT NULL,
GiaTien INT NOT NULL,
)
```

- Bảng đơn giá

```
CREATE TABLE DocGia(
TheDG INT IDENTITY(1,1) PRIMARY KEY,
TenDG NVARCHAR(70) NOT NULL,
)
```

- Bảng phiếu mượn

```
CREATE TABLE PhieuMuon(
MaPM INT PRIMARY KEY,
TheDG INT NOT NULL FOREIGN KEY REFERENCES dbo.DocGia(TheDG),
NgayMuon DATE DEFAULT GETDATE(),
NgayHen DATE NOT NULL,
MaNV VARCHAR(10) FOREIGN KEY REFERENCES dbo.NhanVien(MaNV),
NgayTra DATE NULL,
)
```

- Bảng PMCT



```
CREATE TABLE PMCT(
MaPM INT NOT NULL,
MaSach INT NOT NULL,
TrangThai BIT DEFAULT 0,
PRIMARY KEY (MaPM, MaSach),
CONSTRAINT FK_MaPM FOREIGN KEY (MaPM) REFERENCES PhieuMuon(MaPM),
CONSTRAINT FK_MaSach FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)
)
```

2.1.2. Cấu trúc bảng:


- Bảng Nhân viên:

Column Name	Data Type	Allow Nulls
MaNV	varchar(10)	<input type="checkbox"/>
TenNV	nvarchar(80)	<input type="checkbox"/>
ChucVu	bit	<input checked="" type="checkbox"/>
Email	varchar(60)	<input type="checkbox"/>
TaiKhoan	varchar(50)	<input type="checkbox"/>
MatKhau	varchar(32)	<input type="checkbox"/>

- Bảng chủ đề

	Column Name	Data Type	Allow Nulls
	MaCD	int	<input type="checkbox"/>
	TenCD	nvarchar(80)	<input type="checkbox"/>
			<input type="checkbox"/>


- Bảng Nhà xuất bản

	Column Name	Data Type	Allow Nulls
	MaNXB	int	<input type="checkbox"/>
	TenNXB	nvarchar(80)	<input type="checkbox"/>
	SDT	varchar(15)	<input type="checkbox"/>
	DiaChi	nvarchar(100)	<input type="checkbox"/>

- Bảng tác giả

	Column Name	Data Type	Allow Nulls
	MaTG	int	<input type="checkbox"/>
	TenTG	nvarchar(80)	<input type="checkbox"/>
	GioiTinh	bit	<input checked="" type="checkbox"/>
	NgaySinh	date	<input type="checkbox"/>

- Bảng sách

	Column Name	Data Type	Allow Nulls
	MaSach	int	<input type="checkbox"/>
	TieuDe	nvarchar(80)	<input type="checkbox"/>
	TacGia	varchar(80)	<input type="checkbox"/>
	MaNXB	int	<input type="checkbox"/>
	NamXB	int	<input type="checkbox"/>
	MaCD	int	<input type="checkbox"/>
	NgonNgu	nvarchar(40)	<input type="checkbox"/>
	SoTrang	int	<input type="checkbox"/>
	SoLuong	int	<input type="checkbox"/>
	GiaTien	int	<input type="checkbox"/>

- Bảng đơn giá

	Column Name	Data Type	Allow Nulls
🔑	TheDG	int	<input type="checkbox"/>
	TenDG	nvarchar(70)	<input type="checkbox"/>

- Bảng phiếu mượn

	Column Name	Data Type	Allow Nulls
🔑	MaPM	int	<input type="checkbox"/>
	TheDG	int	<input type="checkbox"/>
	NgayMuon	date	<input checked="" type="checkbox"/>
	NgayHen	date	<input type="checkbox"/>
	MaNV	varchar(10)	<input checked="" type="checkbox"/>
	NgayTra	date	<input checked="" type="checkbox"/>

- Bảng PMCT

	Column Name	Data Type	Allow Nulls
🔑	MaPM	int	<input type="checkbox"/>
🔑	MaSach	int	<input type="checkbox"/>
	TrangThai	bit	<input checked="" type="checkbox"/>

2.1.3. Dữ liệu của mỗi bảng:

- Bảng Nhân viên:

```
INSERT INTO dbo.NhanVien
```

```
(MaNV,TenNV,ChucVu,Email,TaiKhoan,MatKhou)
```

```
VALUES
```

```
('NV01', N'phạm minh tuần', 1,
```

```
'phamminhtuan475@gmail.com','tuan','1')
```

	MaNV	TenNV	ChucVu	Email	TaiKhoan	MatKhou
1	NV01	phạm minh tuần	1	phamminhtuan475@gmail.com	tuan	1

- Bảng chủ đề

```
INSERT INTO ChuDe VALUES (N'thuỷ lợi')
```

```
INSERT INTO ChuDe VALUES (N'Công Nghệ Thông Tin')
```

```
INSERT INTO ChuDe VALUES (N'Đồ Họa')
```

```
INSERT INTO ChuDe VALUES (N'Lập Trình')
```

```
INSERT INTO ChuDe VALUES (N'kinh Tế')
```

```
INSERT INTO ChuDe VALUES (N'Ngoại Ngữ')
```

```
INSERT INTO ChuDe VALUES (N'cấp thoát nước')
```

```
INSERT INTO ChuDe VALUES (N'Đất đai')
```

	MaCD	TenCD
1	7	cấp thoát nước
2	2	Công Nghệ Thông Tin
3	8	Đất đai
4	3	Đồ Họa
5	5	kinh Tế
6	4	Lập Trình
7	6	Ngoại Ngữ
8	1	thủy lợi

- Bảng Nhà xuất bản

```
INSERT INTO NhaXuatBan VALUES (N'Tổng hợp TP.HCM',0000000000,N'Hà Nội')
```

```
INSERT INTO NhaXuatBan VALUES (N'Phụ nữ',0000000000,N'Hà Nội')
```

```
INSERT INTO NhaXuatBan VALUES (N'Kim đồng',0000000000,N'Hà Nội')
```

	MaNXB	TenNXB	SDT	DiaChi
1	1	Tổng hợp TP.HCM	0912345673	Hà Nội
2	2	Phụ nữ	0912345671	Hà Nội
3	3	Kim đồng	0912345662	Hà Nội

- Bảng tác giả

```
INSERT INTO [dbo].[TacGia] VALUES(N'Phan Ý yên',0,'8-21-1992')
```

```
INSERT INTO [dbo].[TacGia] VALUES(N'Hà Thủy Nguyên',0,'10-10-1988')
```

```
INSERT INTO [dbo].[TacGia] VALUES(N'Triệu Yến Quỳnh',0,'3-7-1995')
```

	MaTG	TenTG	GioiTinh	NgaySinh
1	1	Phan Ý yên	0	1992-08-21
2	2	Hà Thủy Nguyên	0	1988-10-10
3	3	Triệu Yến Quỳnh	0	1995-03-07

- Bảng sách

```
INSERT INTO Sach VALUES (N'AQ Chỉ số vượt khó', '1', '1', '2000', '1', N'Tiếng Anh', 600, 250, 300000)
```



```

INSERT INTO Sach VALUES (N'Ai hiểu khách hàng người đó
bán được hàng','2','2','2000','1',N'Tiếng
Anh',600,250,300000)
INSERT INTO Sach VALUES (N'Bài học vô giá từ những điều
bình dị','3','3','2000','1',N'Tiếng Anh',600,250,300000)
INSERT INTO Sach VALUES (N'Bí mật hành trình tình
yêu','4','4','2000','1',N'Tiếng Anh',600,250,300000)
INSERT INTO Sach VALUES (N'Bí mật ngôn ngữ cơ
thể','5','5','2000','1',N'Tiếng Anh',600,250,300000)
INSERT INTO Sach VALUES (N'Bí mật tình yêu Tập
1','7','6','2000','1',N'Tiếng Anh',600,250,300000)
INSERT INTO Sach VALUES (N'Nhật ký son môi', 2, 2, 2013,
1, N'Tiếng việt', 222, 150,120000)
INSERT INTO Sach VALUES (N'Giáo trình phương pháp học
tốt', N'Tô Quang Phiệt', 1, 2012, 2, N'Tiếng việt', 244,
150,150000)
INSERT INTO Sach VALUES (N'Sách tham khảo nấu ăn', N'Nhật
Tiến', 5, 2017, 4, N'Tiếng việt', 1022, 210,220000)
INSERT INTO Sach VALUES (N'Lập trình Java cơ bản', N'Lê
Dư', 2, 2018, 7, N'Tiếng việt', 872, 330,196000)
INSERT INTO Sach VALUES (N'Học Toán siêu tốc', N'Nguyễn
Xuân Hoàng', 4, 2003, 3, N'Tiếng việt', 152, 664,120000)
INSERT INTO Sach VALUES (N'Ôn tập môn Toán 12', N'Nguyễn
Xuân Hoàng', 2, 2013, 3, N'Tiếng việt', 290, 50,160000)
INSERT INTO Sach VALUES (N'Toán lớp 12', N'Tô Quang
Phiệt', 5, 2017, 1, N'Tiếng việt', 872, 321,320000)
INSERT INTO Sach VALUES (N'Lý lớp 12', N'Lê Dư', 5, 2017,
1, N'Tiếng việt', 122, 150,134500)
INSERT INTO Sach VALUES (N'Hóa lớp 12', N'Hữu Mai', 5,
2017, 1, N'Tiếng việt', 754, 230,254000)

```

```

INSERT INTO Sach VALUES (N'Văn lớp 12', N'Bình Nguyễn
Lộc', 5, 2017, 1, N'Tiếng việt', 264, 320,563000)
INSERT INTO Sach VALUES (N'Lập trình Java nâng cao', N'Hà
Thủy Nguyễn', 2, 2018, 9, N'Tiếng việt', 233, 50,220000)
INSERT INTO Sach VALUES (N'Thiết kế đồ họa cơ bản', N'Du
Phong', 12, 2012, 8, N'Tiếng việt', 124, 650,215000)
INSERT INTO Sach VALUES (N'Thiết kế đồ họa nâng cao',
N'Cổ Mạn', 5, 2003, 8, N'Tiếng việt', 22, 110,160000)
INSERT INTO Sach VALUES (N'Cách làm giàu từ kinh doanh',
N'Gào', 3, 2015, 11, N'Tiếng việt', 266, 150,120000)
INSERT INTO Sach VALUES (N'Lập trình SQL', N'Minh Mẫn',
7, 2013, 9, N'Tiếng việt', 265, 250,322000)
INSERT INTO Sach VALUES (N'Thiết kế đồ họa theo chủ đề',
N'Đồng Hoa', 8, 2012, 1, N'Tiếng việt', 154, 140,190000)
INSERT INTO Sach VALUES (N'Ngôn ngữ Nga', N'Tuệ Nghi',
11, 2013, 12, N'Nga', 263, 112,266000)
INSERT INTO Sach VALUES (N'Bổ trợ kiến thức văn học',
N'Lê Dư', 12, 2011, 5, N'Tiếng việt', 166, 230,175000)
INSERT INTO Sach VALUES (N'Ảnh hưởng của công nghệ thông
tin', N'Lê Văn Trương', 3, 2019, 7, N'Tiếng việt', 166,
150,432000)
INSERT INTO Sach VALUES (N'Siêu lập trình Java', N'Đồng
Hoa', 12, 2015, 9, N'Tiếng việt', 122, 320,532000)
INSERT INTO Sach VALUES (N'Giáo trình lớp 11', N'Lê Dư',
4, 2013, 2, N'Tiếng việt', 112, 50,120000)
INSERT INTO Sach VALUES (N'Toán lớp 12', N'Triệu yến
Quỳnh', 4, 2013, 1, N'Tiếng việt', 552, 350,176000)
INSERT INTO Sach VALUES (N'Toán học lớp 6', N'Minh Anh',
2, 2013, 1, N'Tiếng việt', 4212, 150,190000)

```

```

INSERT INTO Sach VALUES (N'Lập trình C++', N'Minh Hùng',
1, 2013, 9, N'Tiếng việt', 400, 400,155000)
INSERT INTO Sach VALUES (N'Cách làm giàu', N'Lục Nguyên',
7, 2013, 11, N'Tiếng việt', 343, 170,390000)
INSERT INTO Sach VALUES (N'Bổ trợ kiến thức giao tiếp',
N'Lê Ái', 8, 2013, 5, N'Tiếng Anh', 2122, 150,1520000)
INSERT INTO Sach VALUES (N'Học toán cùng công nghệ',
N'Mai Tuệ Vân', 3, 2013, 1, N'Tiếng việt', 332,
12,210000)
INSERT INTO Sach VALUES (N'Văn học lớp 11', N'Anh Khang',
6, 2013, 1, N'Tiếng việt', 1262, 150,620000)
INSERT INTO Sach VALUES (N'Hóa học lớp 10', N'Nguyễn
Phong Việt', 4, 2013, 1, N'Tiếng việt', 238, 321,156000)
INSERT INTO Sach VALUES (N'Sinh học lớp 8', N'Minh Anh',
2, 2013, 1, N'Tiếng việt', 234, 642,235000)

```

1	1	AQ Chỉ số vượt khó	1	1	2000	1	Tiếng Anh	600	250	300000
2	2	Ai hiểu khách hàng người đó bán được hàng	2	2	2000	1	Tiếng Anh	600	250	300000
3	3	Bài học vô giá từ những điều bình dị	3	3	2000	1	Tiếng Anh	600	250	300000
4	4	Bí mật hành trình tình yêu	4	4	2000	1	Tiếng Anh	600	250	300000
5	5	Bí mật ngôn ngữ cơ thể	5	5	2000	1	Tiếng Anh	600	250	300000
6	6	Bí mật tình yêu Tập 1	7	6	2000	1	Tiếng Anh	600	250	300000
7	7	Nhật ký son môi	2	2	2013	1	Tiếng việt	222	150	120000
8	8	Giáo trình phương pháp học tốt	9	1	2012	2	Tiếng việt	244	150	150000
9	9	Sách tham khảo nấu ăn	23	5	2017	4	Tiếng việt	1022	210	220000
10	10	Lập trình Java cơ bản	14	2	2018	7	Tiếng việt	872	330	196000

- Bảng đơn giá

```
INSERT INTO dbo.DocGia
```

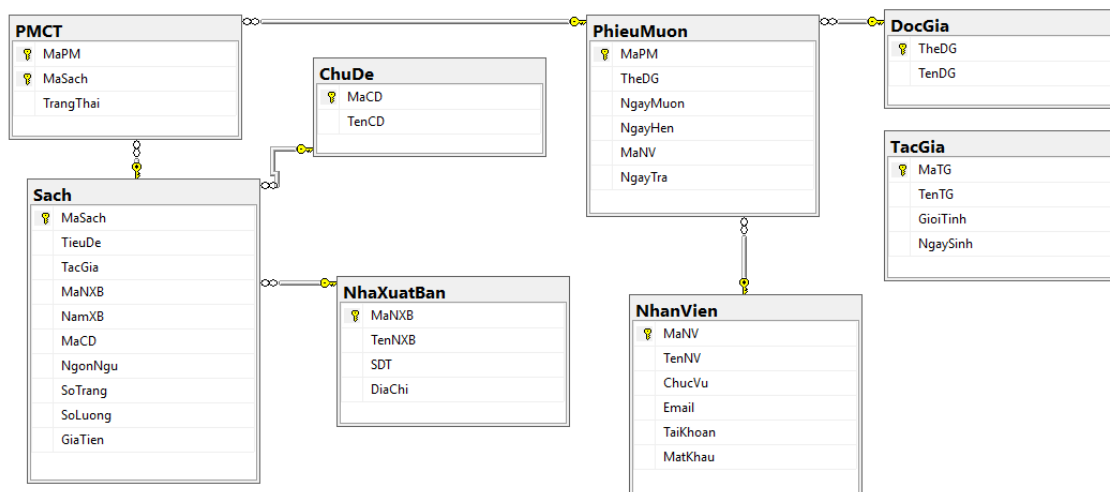
```
(TenDG)VALUES
```

```
(N'phạm minh tuần'),(N'hà chí bảo '), (N'nguyễn mạnh hùng
vĩ')
```

	TheDG	TenDG
1	1	phạm minh tuần
2	2	hà chí bảo
3	3	nguyễn mạnh hùng vĩ

- Bảng phiếu mượn
- Bảng PMCT

2.1.4. Màn hình Diagram liên kết các bảng:



2.2. Thao tác trên CSDL trên HQT CSDL:

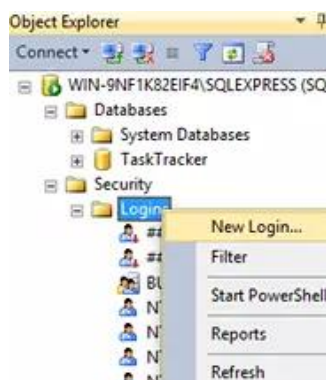
2.2.1. Tạo và xóa tài khoản người dùng đăng nhập quyền sql server + Bằng công cụ:

User Logins: SQL Server cho phép người dùng có thể tạo tài khoản và đăng nhập vào. Mỗi người dùng đều truy cập vào trong SQL Server qua 1 tài khoản riêng. Khi tạo 1 tài khoản người dùng mới, quản trị viên sẽ cấp cho tài khoản này các quyền(Server Roles) và phạm vi có thể truy cập(Database Schemas).

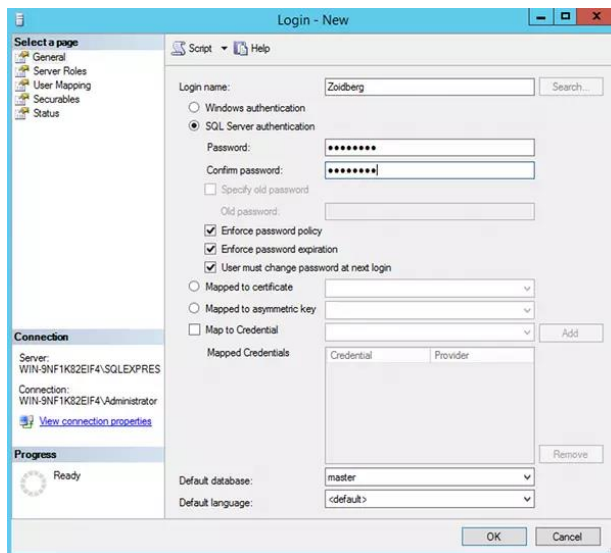
Tạo tài khoản bằng công cụ:

B1: Sử dụng SQL Server Management Studio, chọn dấu + để mở rộng tùy chọn **Security** và chuột phải vào **Logins**.

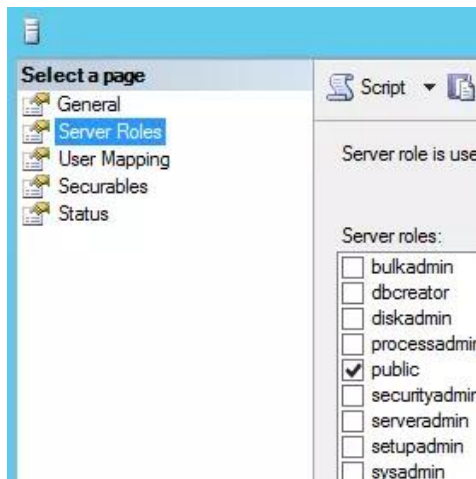
B2: Chọn **New Login...**



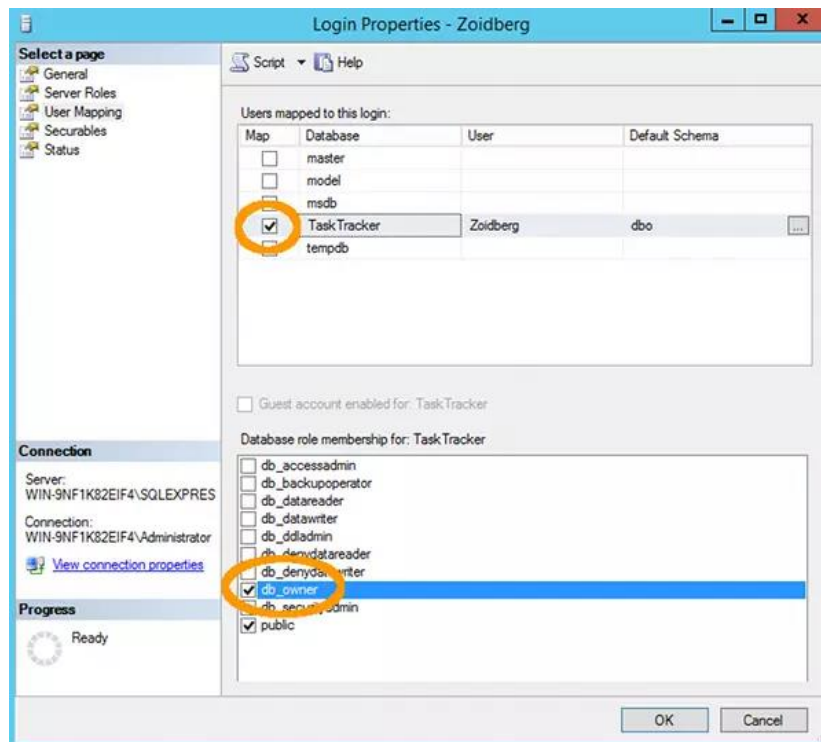
B3: Tiếp đó cần điền đầy đủ các thông tin trong tab **General**.



B4: Sau đó vào tab **Server Roles** để chọn các đặc quyền cho tài khoản trong SQL Server.



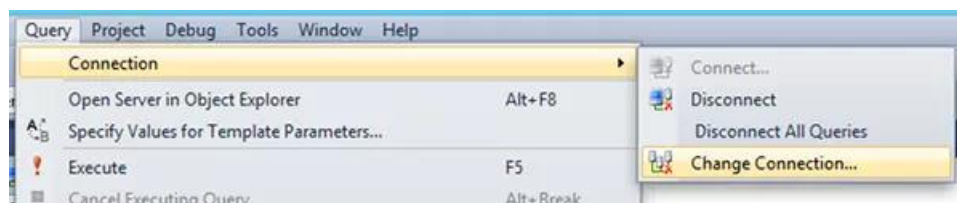
B5: Tiếp đến là tab **User Mapping** để chọn các DataBase mà tài khoản có thể kết nối tới, Schema bên trong Database. Sau đó chọn các đặc quyền(Role) đối với Database. *Note: Role ở phần này phạm vi ảnh hưởng là trên Database đã chọn. Role ở phần 4 có ảnh hưởng trên SQL Server.



B6: Chọn **OK** để hoàn thành việc tạo tài khoản.

* Đăng nhập thử tài khoản mới:

B1: Kết nối lại với SQL Server với tài khoản mới chúng ta vừa tạo bằng cách chọn **Change Connection...** bên trong menu **Query** của toolbar.



B2: Sau đó sẽ có form hiện ra để nhập các thông tin chúng ta vừa đăng ký phía trên.

* Tạo tài khoản bằng lệnh:

Mô tả: **CREATE USER** tạo cơ sở dữ liệu user để đăng nhập vào SQL Server. Một cơ sở dữ liệu user sẽ được ánh xạ đến Login, định danh được dùng để kết nối với một instance SQL Server cụ thể.

Cú pháp: `CREATE USER user_name FOR LOGIN login_name;`

VD:

2.2.2. Viết câu lệnh Insert:

Mô tả: Lệnh **INSERT** dùng để thêm mới dữ liệu vào bảng (Table).

DEPARTMENTS				
DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID	
10	Administration	200	1700	
20	Marketing	201	1800	
30	Purchasing	114	1700	
40	Human Resources	203	2400	
50	Shipping	121	1500	
60	IT	103	1400	
70	Public Relations	204	2700	

Dòng mới →

80

Sales

145

2500

↑ Thêm vào

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID	
10	Administration	200	1700	
20	Marketing	201	1800	
30	Purchasing	114	1700	
40	Human Resources	203	2400	
50	Shipping	121	1500	
60	IT	103	1400	
70	Public Relations	204	2700	
80	Sales	145	2500	

Cú pháp:

INSERT Tên_bảng(Cột_1, Cột_2,... Cột_n)

VALUES(Giá_trị_1, Giá_trị_2,... Giá_trị_n)

--Hoặc (có thể thêm INTO):

INSERT INTO Tên_bảng(Cột_1, Cột_2,... Cột_n)

VALUES(Giá_trị_1, Giá_trị_2,... Giá_trị_n)

VD:

INSERT INTO dbo.DocGia

(TenDG)**VALUES**

(N'phạm minh tuần'),(N'hà chí bảo '),(N'nguyễn mạnh hùng vĩ')

	TheDG	TenDG
1	1	phạm minh tuần
2	2	hà chí bảo
3	3	nguyễn mạnh hùng vĩ

2.2.3. Viết câu lệnh update:

Mô tả: Lệnh **UPDATE** dùng để sửa dữ liệu hiện có trong bảng (Table). Có thể sửa nhiều dòng tại một thời điểm.

Cú pháp:

Thông thường: (điều kiện từ 1 bảng)

UPDATE Tên_bảng

```

SET Cột_1 = Giá_trị_1, Cột_2 = Giá_trị_2,... Cột_n =
Giá_trị_n
WHERE Điều_kiện
Lệnh Update...from: (điều kiện kết hợp từ nhiều bảng)
UPDATE Tên_bảng
SET Tên_cột = Biểu_thức,...
FROM Tên_bảng_1
INNER|LEFT|RIGHT JOIN Tên_bảng_2 ON Biểu_thức_liên_kết
WHERE Điều_kiện_sửa_đổi

```

VD:

```
UPDATE [dbo].[Sach] SET TacGia=12 WHERE MaSach=1
```

MaSach	TieuDe	TacGia
1	AQ Chỉ số vượt khó	10



MaSach	TieuDe	TacGia
1	AQ Chỉ số vượt khó	12

2.3. Store Procedure, Funtion, Trigger:

2.3.1. Store Procedure:

- Định nghĩa:

- Là đoạn chương trình kịch bản (programming scripts) với các câu lệnh SQL nhúng (embedded SQL) được lưu dưới dạng đã được biên dịch và thi hành thực tiếp bởi MySQL server,
- SP cho phép lưu trữ các logic ứng dụng trên CSDL. Khi gọi SP lần đầu tiên, MySQL sẽ tạo một lịch thực thi và lưu trữ nó trong bộ nhớ đệm. Ở những lần gọi hàm tiếp theo, MySQL sử dụng sử dụng lại lịch thực thi được lưu rất nhanh với hiệu xuất đáng tin cậy.
- SP là một mã SQL đã chuẩn bị sẵn có thể lưu, do đó đoạn mã có thể được sử dụng lại nhiều lần. Vì vậy, hãy lưu các lệnh truy vấn dưới dạng một thủ tục được lưu trữ, sau đó chỉ cần gọi nó để thực thi nó.

- Cũng có thể chuyển các tham số cho một thủ tục được lưu trữ, để thủ tục được lưu trữ có thể hoạt động dựa trên (các) giá trị tham số được truyền vào.

- Ưu điểm và nhược điểm của Stored Procedure:

STORED PROCEDURE	Ưu điểm	Nhược điểm
	Hiệu suất tốt: các cuộc gọi thủ tục nhanh chóng và hiệu quả vì các thủ tục lưu trữ được biên dịch một lần và được lưu trữ ở dạng thực thi. Mã thực thi được tự động lưu trữ, do đó làm giảm yêu cầu bộ nhớ.	Khả năng kiểm tra: logic nghiệp vụ được gói gọn trong các thủ tục lưu trữ nên rất khó kiểm tra (nếu được kiểm tra).
	Năng suất cao hơn: vì khả năng tái sử dụng mà không cần phải viết lại các câu lệnh SQL lặp đi lặp lại nên do đó năng suất tạo ra cao hơn.	Khả năng gỡ lỗi: tùy thuộc vào các hệ quản trị cơ sở dữ liệu, việc gỡ lỗi các thủ tục sẽ không thực hiện được hoặc cực kỳ khó hiểu.
	Khả năng mở rộng và bảo trì: các thủ tục lưu trữ tăng khả năng mở rộng bằng cách cách ly xử lý ứng dụng trên server. Nếu cần bất kỳ thay đổi nào, bạn chỉ cần thay đổi thủ tục lưu trữ mà không cần chạm vào mã ứng dụng.	Sợ thay đổi: một trong những nhược điểm lớn nhất của các thủ tục lưu trữ là cực kỳ khó để biết phần nào của hệ thống sử dụng chúng và phần nào không.
	Bảo mật tốt: ta có thể viết một stored procedure để truy xuất dữ liệu và chỉ cho phép user đó được sử dụng stored procedure đã viết sẵn mà thôi chứ không thể "đụng" đến các tables đó một cách trực tiếp. Ngoài ra stored procedure có	

	thể được encrypt (mã hóa) để tăng cường tính bảo mật.	
--	-------------------------------------------------------	--

- Cú pháp:

Để tạo thủ tục lưu trữ trong SQL Server, bạn có thể sử dụng cú pháp sau:

```
CREATE { PROCEDURE | PROC } [schema_name.]procedure_name
[ @parameter [type_schema_name.] datatype
[ VARYING ] [ = default ] [ OUT | OUTPUT | READONLY ]
, @parameter [type_schema_name.] datatype
[ VARYING ] [ = default ] [ OUT | OUTPUT | READONLY ] ]

[ WITH { ENCRYPTION | RECOMPILE | EXECUTE AS Clause } ]
[ FOR REPLICATION ]
```

AS

BEGIN

[declaration_section]

executable_section

END;

Tham số:

- *schema_name*: Tên schema (lược đồ) sở hữu procedure.
- *procedure_name*: Tên gán cho procedure
- *@parameter*: Một hay nhiều tham số được truyền vào hàm.
- *type_schema_name*: Kiểu dữ liệu của schema (nếu có).
- *Datatype*: Kiểu dữ liệu cho @parameter.
- *Default*: Giá trị mặc định gán cho @parameter.
- *OUT/OUTPUT*: @parameter là một tham số đầu ra
- *READONLY*: @parameter không thể bị procedure ghi đè lên.
- *ENCRYPTION*: Mã nguồn (source) của procedure sẽ không được lưu trữ dưới dạng text trong hệ thống.

- *RECOMPILE*: Truy vấn sẽ không được lưu ở bộ nhớ đệm (cache) cho thủ tục này.
- *EXECUTE AS clause*: Xác định ngữ cảnh bảo mật để thực thi thủ tục.
- *FOR REPLICATION*: Procedure đã lưu sẽ chỉ được thực thi trong quá trình replication (nhân bản).

Sau đó ta có thể gọi thủ tục được lưu trữ bằng cú pháp sau:

`EXEC procedure_name parameter1_value, parameter2_value, ...`

Khi tạo các thủ tục được lưu trữ, điều quan trọng là phải tuân theo các phương pháp hay nhất như tham số hóa đầu vào, xử lý lỗi đúng cách và hạn chế quyền truy cập vào dữ liệu nhạy cảm.

- Các loại SP, ưu và khuyết của từng loại,...

Các loại SP	Ưu điểm	Nhược điểm
System Stored Procedure	Được tạo sẵn trong hệ thống, cung cấp các chức năng phổ biến và tiện lợi cho người dùng	Không thể sửa đổi hoặc thay đổi nội dung của hệ thống lưu trữ các thủ tục được lưu trữ này
Custom Stored Procedure	Cung cấp các chức năng tùy chỉnh theo yêu cầu của người dùng, giúp tối ưu hóa hiệu suất và đơn giản hóa việc phát triển ứng dụng	Cần phải tạo và quản lý các thủ tục được lưu trữ này, yêu cầu sự kiểm tra và quản lý chặt chẽ để đảm bảo tính toàn vẹn và toàn bộ dữ liệu
Query Stored Procedure	Cho phép thực hiện các truy vấn phức tạp, thao tác với nhiều bảng và tối ưu hóa hiệu suất	Cần sử dụng ngôn ngữ SQL để tạo truy vấn, có thể gây khó khăn cho những người mới bắt đầu với SQL
Data Manipulation Stored Procedure	Cho phép thực hiện các thao tác bổ sung, chỉnh sửa, xóa dữ liệu trên cơ sở dữ liệu	Cần phải kiểm tra và quản lý chặt chẽ để đảm bảo tính toàn vẹn và

	sở dữ liệu	toàn bộ dữ liệu.
Xử Lý Sự Kiện Stored Procedure	Cho phép xử lý các sự kiện và thông báo trong hệ thống, đảm bảo tính liên tục và ổn định của ứng dụng.	Yêu cầu các kỹ năng đặc biệt để tạo và quản lý các thủ tục được lưu trữ này, đồng thời phải kiểm tra và quản lý chặt chẽ để đảm bảo tính toàn vẹn và an toàn dữ liệu.

➤ **Store procedure trên CSDL**

```
CREATE PROC sp_thong_ke_sach_theo_chu_de
AS BEGIN
    SELECT TenCD,
           COUNT(Sach.MaCD) 'so luong sach'
    FROM Sach
    INNER JOIN ChuDe ON ChuDe.MaCD = Sach.MaCD
    GROUP BY Sach.MaCD, TenCD
END
GO
```

2.3.2. Function:

- Định nghĩa:

- **Function (Hàm)** là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh được nhóm lại với nhau và được tạo ra với mục đích sử dụng lại. Trong SQL Server, hàm được lưu trữ và bạn có thể truyền các tham số vào cũng như trả về các giá trị.

- Gồm 2 loại: Function hệ thống và Function do người dùng tự định nghĩa.

Trong đó: Function người dùng tự định nghĩa gồm 2 loại:

- + Scalar-valued: Trả về giá trị vô hướng của các kiểu dữ liệu T-SQL.
- + Table-valued: Trả về bảng, là kết quả của một hoặc nhiều lệnh.

- Cú pháp:

- Tạo Function trả về giá trị Scalar-valued

```

CREATE FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>],
..., [...]])
RETURNS <kiểu dữ liệu>
[WITH ENCRYPTION]
[AS]
BEGIN
    [Thân của hàm]
    RETURN <Biểu thức giá trị đơn>
END

```

Trong đó:

- + Tên function: Tên của hàm chúng ta sẽ tạo
- + Tên tham số: Là các tham số Input cho hàm. Khai báo bao gồm tên của tham số (trước tên tham số sử dụng tiền tố @), kiểu dữ liệu của tham số, chúng ta có thể chỉ định giá trị mặc định cho tham số. Có thể chỉ định nhiều tham số đầu vào
 - + RETURNS: từ khóa này chỉ định kiểu dữ liệu hàm sẽ trả về. Kiểu dữ liệu phải được chỉ định kiểu độ dài dữ liệu. Ví dụ: varchar(100)
 - + WITH ENCRYPTION: Từ khóa chỉ định code của hàm sẽ được mã hóa trong bảng syscomments.
 - + AS: Từ khóa cho biết code của hàm bắt đầu.
 - + BEGIN: Đi cùng với END để tạo thành bao khối bao các câu lệnh trong thân hàm.
 - + RETURN: Từ khóa này sẽ gửi giá trị tới thủ tục gọi hàm.
- Tạo Function trả về giá trị Table-valued:
 - Function Table-valued có 2 loại:
 - Hàm giá trị bảng đơn giản: Trả về bảng, là kết quả của một câu lệnh SELECT đơn

```

CREATE FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>],
..., [...]])
RETURNS TABLE

```

[WITH ENCRYPTION]

[AS]

RETURN <Câu lệnh SQL>

END

- Hàm giá trị bảng đa câu lệnh: Trả về bảng, là kết quả của nhiều câu lệnh.

CREATE FUNCTION <Tên function>

([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], ..., [...]])

RETURNS @<tên biến trả về> TABLE (<tên cột 1> <kiểu dữ liệu> [tùy chọn thuộc tính], ..., <tên cột n> <kiểu dữ liệu> [tùy chọn thuộc tính])

[AS]

BEGIN

<Câu lệnh SQL>

RETURN

END

- Ưu điểm và nhược điểm:

Ưu điểm	Nhược điểm
Tối ưu hóa hiệu hiệu: Chức năng có thể được sử dụng để tối ưu hóa hiệu hiệu truy vấn bằng cách thực hiện các tính toán và tính toán trước khi trả về kết quả cho truy vấn	Khó sửa đổi: Nếu chức năng được sử dụng nhiều trong truy vấn, việc sửa đổi nó có thể ảnh hưởng đến các truy vấn khác.
Tính linh hoạt: Chức năng cho phép tạo các tính năng tùy chỉnh và thực hiện các tính toán phức tạp	Ảnh hưởng đến hiệu suất: Chức năng có thể ảnh hưởng đến hiệu suất truy vấn nếu chúng được sử dụng không đúng cách hoặc không được tối ưu hóa đúng cách
Dễ sử dụng: Chức năng có thể được sử	Khó theo dõi và quản lý: Việc quản lý

dùng trong truy vấn SQL như một phần của biểu thức hoặc câu lệnh Select	và theo dõi các chức năng có thể trở nên khó khăn khi chúng được sử dụng trong nhiều truy vấn
-------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------

➤ **Function trên CSDL**

```

CREATE FUNCTION fn_thong_ke_tac_gia (
    @tentg NVARCHAR(80),
    @a VARCHAR(MAX),
    @masach INT
)
RETURNS INT
AS BEGIN
    DECLARE @matg INT
    SET @matg = (SELECT MaTG FROM TacGia WHERE TenTG =
    @tentg)

    DECLARE @b INT

    WHILE CHARINDEX(',', @a) != 0
        BEGIN
            SET @b = LTRIM(RTRIM(SUBSTRING(@a, 0,
CHARINDEX(',', @a))))
            SET @a = LTRIM(RTRIM(SUBSTRING(@a,
CHARINDEX(',', @a) + 1, LEN(@a))))
            IF @b = @matg
                BEGIN
                    RETURN (SELECT @masach)
                    BREAK
                END
        END

    END

    IF LTRIM(RTRIM(@a)) = @matg

```

```
BEGIN
    RETURN (SELECT @masach)
END
```

```
RETURN NULL
END
```

```
go
```

2.3.3. Trigger:

- Định nghĩa:

- Trigger là một thủ tục SQL được thực thi ở phía sau máy chủ khi có một số sự kiện như Chèn, Xóa, hoặc Cập nhật. **Trigger** là một loại thủ tục lưu sẵn đặc biệt (không có tham số) được thực thi (thực thi) một cách tự động khi có một sự kiện thay đổi dữ liệu (sửa đổi dữ liệu). Trigger được lưu trữ và quản lý trong Server DB, được sử dụng trong trường hợp ta muốn kiểm tra các ràng buộc toàn vẹn trong DB.
- Trigger thường được sử dụng để kiểm tra ràng buộc (kiểm tra ràng buộc) trên nhiều quan hệ (nhiều bảng/bảng) hoặc trên nhiều dòng (nhiều bản ghi) của bảng.
- Ngoài ra việc sử dụng Trigger để chương trình có những hàm chạy ngầm nhằm phục vụ nhưng trường hợp hữu hạn và thường không sử dụng cho mục đích kinh doanh hoặc giao dịch.
- Ngăn chặn việc xóa những dữ liệu quan trọng. (có thể sử dụng sao lưu các bảng dữ liệu quan trọng khác trong phòng khi ...bị xóa bên ngoài ý muốn).
- Có 3 loại Trigger như sau:
 - **Trigger cho DML** (ngôn ngữ thao tác dữ liệu): bao gồm 3 kiểu là INSERT, UPDATE và DELETE.
 - **Trigger cho DDL** (ngôn ngữ định nghĩa dữ liệu): sử dụng cho các lệnh CREATE, ALTER và DROP.
 - **Trigger cho LOGON**: sử dụng để kiểm soát các lệnh khi đăng nhập vào máy chủ.

- Cú pháp:

```
CREATE TRIGGER tên_trigger  
ON { Tên_bảng }  
[ WITH <Options> ]  
{ FOR | AFTER | INSTEAD OF }  
{ [INSERT – chèn], [UPDATE – Cập nhật] , [DELETE – Xóa]}
```

- Ưu và nhược điểm của Trigger:

Ưu điểm	Nhược điểm
Trình Trigger có thể bắt lỗi logic nghiệp vụ ở mức csdl.	Trình Trigger chỉ là một phần mở rộng của hợp lệ kiểm tra tính năng của dữ liệu chứ không phải thay thế được hoàn thành toàn bộ công việc này.
Có thể sử dụng trình Trigger theo một cách khác để thay thế việc thực hiện các công việc đã hẹn giờ theo lịch.	Trigger hoạt động tạm dừng trong csdl, không hiển thị ở tầng giao diện. Do đó, khó chỉ ra điều gì xảy ra ở tầng csdl
Trigger rất hiệu quả khi được sử dụng để kiểm tra những thay đổi của dữ liệu trong bảng	Trình Trigger thực hiện cập nhật lên bảng dữ liệu vì thế nó làm tăng lượng công việc lên csdl và làm cho hệ thống chạy chậm lại.

➤ Trigger trên CSDL

```
create trigger tg_tra_sach on PhieuMuon for insert
```

```
as if (select NgayMuon from inserted) < (select NgayHen from inserted)
```

```
begin
```

```
print N'Phải trả sách trước ngày hẹn'
```

```
rollback transaction
```

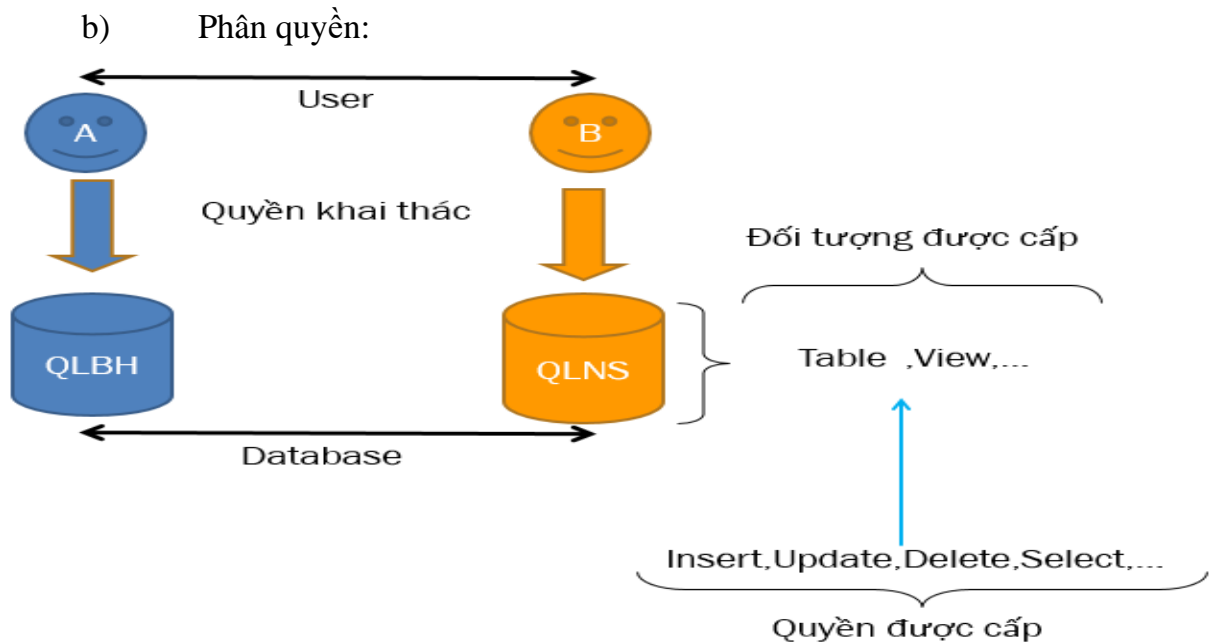
end

2.4. Bảo mật, phân quyền:

a) Khái niệm:

- Bảo mật là một trong những yếu tố đóng vai trò quan trọng đối với sự sống còn của cơ sở dữ liệu. Hầu hết các hệ quản trị cơ sở dữ liệu thương mại hiện nay đều cung cấp khả năng bảo mật cơ sở dữ liệu với những chức năng như:
 - Cấp phát quyền truy cập cơ sở dữ liệu cho người dùng và các nhóm người dùng, phát hiện và ngăn chặn những thao tác trái phép của người sử dụng trên cơ sở dữ liệu.
 - Cấp phát quyền sử dụng các câu lệnh, các đối tượng cơ sở dữ liệu đối với người dùng.
 - Thu hồi (huỷ bỏ) quyền của người dùng.
- Bảo mật dữ liệu trong SQL được thực hiện dựa trên ba khái niệm chính sau đây:
 - Người dùng cơ sở dữ liệu (Database user): Là đối tượng sử dụng cơ sở dữ liệu, thực thi các thao tác trên cơ sở dữ liệu như tạo bảng, truy xuất dữ liệu,... Mỗi một người dùng trong cơ sở dữ liệu được xác định thông qua tên người dùng (User ID). Một tập nhiều người dùng có thể được tổ chức trong một nhóm và được gọi là nhóm người dùng (User Group). Chính sách bảo mật cơ sở dữ liệu có thể được áp dụng cho mỗi người dùng hoặc cho các nhóm người dùng.
 - Các đối tượng cơ sở dữ liệu (Database objects): Tập hợp các đối tượng, các cấu trúc lưu trữ được sử dụng trong cơ sở dữ liệu như bảng, khung nhìn, thủ tục, hàm được gọi là các đối tượng cơ sở dữ liệu. Đây là những đối tượng cần được bảo vệ trong chính sách bảo mật của cơ sở dữ liệu.
 - Đặc quyền (Privileges): Là tập những thao tác được cấp phát cho người dùng trên các đối tượng cơ sở dữ liệu. Chẳng hạn một người dùng có thể truy xuất dữ liệu trên một bảng bằng câu lệnh SELECT nhưng có thể không thể thực hiện các câu lệnh INSERT, UPDATE hay DELETE trên bảng đó.

- SQL cung cấp hai câu lệnh cho phép chúng ta thiết lập các chính sách bảo mật trong cơ sở dữ liệu:
 - Lệnh GRANT: Sử dụng để cấp phát quyền cho người sử dụng trên các đối tượng cơ sở dữ liệu hoặc quyền sử dụng các câu lệnh SQL trong cơ sở dữ liệu.
 - Lệnh REVOKE: Được sử dụng để thu hồi quyền đối với người sử dụng.



Hình 12. Bảng phân quyền

Câu lệnh GRANT được sử dụng để phân quyền cho người dùng hay nhóm người dùng trên các đối tượng cơ sở dữ liệu. Câu lệnh này thường được sử dụng trong các trường hợp sau:

- Người sở hữu đối tượng cơ sở dữ liệu muốn cho phép người dùng khác quyền sử dụng những đối tượng mà anh ta đang sở hữu.
- Người sở hữu cơ sở dữ liệu phân quyền thực thi các câu lệnh (như CREATE TABLE, CREATE VIEW,...) cho những người dùng khác.

- Phân quyền cho người dùng trên các đối tượng cơ sở dữ liệu:

Chỉ có người sở hữu cơ sở dữ liệu hoặc người sở hữu đối tượng cơ sở dữ liệu mới có thể phân quyền cho người dùng trên các đối tượng cơ sở dữ liệu. Câu lệnh GRANT trong trường hợp này có cú pháp như sau:

GRANT ALL [PRIVILEGES] | các_quyền_cấp_phát

```
[ danh_sách_cột ] ON tên_bảng | tên_khung_nhìn
|ON tên_bảng | tên_khung_nhìn [(danh_sách_cột)]
|ON tên_thủ_tục
|ON tên_hàm
TO danh_sách_người_dùng | nhóm_người_dùng
[WITH GRANT OPTION ]
```

- Phân quyền thực thi các câu lệnh:

Ngoài chức năng cấp phát quyền cho người sử dụng trên các đối tượng cơ sở dữ liệu, câu lệnh GRANT còn có thể sử dụng để cấp phát cho người sử dụng một số quyền trên hệ quản trị cơ sở dữ liệu hoặc cơ sở dữ liệu. Những quyền có thể cấp phát trong trường hợp này bao gồm:

- Tạo cơ sở dữ liệu: CREATE DATABASE.
- Tạo bảng: CREATE TABLE
- Tạo khung nhìn: CREATE VIEW
- Tạo thủ tục lưu trữ: CREATE PROCEDURE
- Tạo hàm: CREATE FUNCTION
- Sao lưu cơ sở dữ liệu: BACKUP DATABASE

Câu lệnh GRANT sử dụng trong trường hợp này có cú pháp như sau:

```
GRANT ALL | danh_sách_câu_lệnh
TO danh_sách_người_dùng
```

c) Thu hồi quyền:

Câu lệnh REVOKE được sử dụng để thu hồi quyền đã được cấp phát cho người dùng. Tương ứng với câu lệnh GRANT, câu lệnh REVOKE được sử dụng trong hai trường hợp:

Thu hồi quyền đã cấp phát cho người dùng trên các đối tượng cơ sở dữ liệu.

Thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu đã cấp phát cho người dùng.

- Thu hồi quyền trên đối tượng cơ sở dữ liệu:

Cú pháp câu lệnh REVOKE sử dụng để thu hồi quyền đã cấp phát trên đối tượng cơ sở dữ liệu có cú pháp như sau:

```
REVOKE [GRANT OPTION FOR]
```

```

ALL [PRIVILEGES] | các_quyền_cần_thu_hồi
[(danh_sách_cột)] ON tên_bảng | tên_khung_nhìn
|ON tên_bảng | tên_khung_nhìn [(danh_sách_cột)]

|ON tên_thủ_tục
|ON tên_hàm
FROM danh_sách_người_dùng
[CASCADE]

```

Câu lệnh REVOKE có thể sử dụng để thu hồi một số quyền đã cấp phát cho người dùng hoặc là thu hồi tất cả các quyền (ALL PRIVILEGES).

- Thu hồi quyền thực thi các câu lệnh:

Việc thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu (CREATE DATABASE, CREATE TABLE, CREATE VIEW,...) được thực hiện đơn giản với câu lệnh REVOKE có cú pháp:

```

REVOKE ALL | các_câu_lệnh_cần_thu_hồi
FROM danh_sách_người_dùng

```

Chương 3: KẾT LUẬN

3.1. Những kết quả đạt được của đồ án:

- Xây dựng được hệ quản trị cơ sở dữ liệu quản lí thư viện
- Ứng dụng Store Procedure, Funtion, Trigger để xử lí dữ liệu.
- Ứng dụng được các thao tác delete, insert, update, create

3.2. Nhược điểm của đồ án:

- xử lý câu lệnh trong Store Procedure, Funtion, Trigger, phân quyền chưa được tối ưu do thời gian còn hạn hẹp và trình độ còn non kém

PHỤ LỤC

HỆ THỐNG QUẢN LÝ THƯ VIỆN Xin Chào : Admin

Quản Lý Sách

Độc Giả

Mượn Trả Sách

Quản Lý Nhân Viên

Báo cáo - Thống Kê

Giới Thiệu

Quản Lý Sách

SÁCH LOẠI SÁCH TÁC GIẢ NXB

Tìm Kiếm

☐ Tên Sách ☐ Loại Sách ☐ Nhà Xuất Bản ☐ Tác Giả

@ 2023 - Đại Học Tài Nguyên và Môi Trường

Giao diện trang chủ

HỆ THỐNG QUẢN LÝ THƯ VIỆN Xin Chào : Admin

Quản Lý Sách

Độc Giả

Mượn Trả Sách

Quản Lý Nhân Viên

Báo cáo - Thống Kê

Giới Thiệu

Quản Lý Sinh Viên - Độc Giả

Tìm Kiếm Sinh Viên

☐ Mã Sinh Viên ☐ Tên Sinh Viên

Mã Sinh Viên

Họ và Tên

Ngành Học

Khoá Học

Số Điện Thoại

Thêm Sửa Xoá

Ghi Huỷ bỏ

Giao diện quản lý sinh viên

HỆ THỐNG QUẢN LÝ THƯ VIỆN

Xin Chào : Admin

Quản Lý Sách

Độc Giả

Mượn Trả Sách

Quản Lý Nhân Viên

Báo cáo - Thống Kê

Giới Thiệu

Quản Lý Sách

SÁCH

LOẠI SÁCH

TÁC GIẢ

NXB

Tìm Kiếm

Tên Sách

Loại Sách

Nhà Xuất Bản

Tác Giả

@ 2023 - Đại Học Tài Nguyên và Môi Trường

Giao diện tra cứu sách

HỆ THỐNG QUẢN LÝ THƯ VIỆN

SÁCH

Mã Sách

Tên Sách

Tác Giả

Nhà Xuất Bản

Loại Sách

Số Trang

Giá Bán

Số Lượng

Thêm

Sửa

Xoá

Ghi

Hủy bỏ

Trang Chủ

Tìm Kiếm

Giao diện quản lý sách

HỆ THỐNG QUẢN LÝ THƯ VIỆN

TÁC GIẢ

Tìm Kiếm

Mã Tác Giả

Tên Tác Giả

Ghi Chú

Thêm

Sửa

Xoá

Ghi

Hủy bỏ

Trang Chủ

Giao diện quản lý tác giả

HỆ THỐNG QUẢN LÝ THƯ VIỆN

LOẠI SÁCH

Tìm Kiếm

Mã Loại

Loại Sách

Ghi Chú

Thêm

Sửa

Xoá

Ghi

Hủy bỏ

Trang Chủ

Giao diện quản lý loại sách

HỆ THỐNG QUẢN LÝ THƯ VIỆN

NHÀ XUẤT BẢN

Tìm Kiếm

Mã Nhà XB

Tên Nhà XB

Ghi Chú

Thêm

Sửa

Xoá

Ghi

Hủy bỏ

Trang Chủ

Giao diện quản lý nhà xuất bản

HỆ THỐNG QUẢN LÝ THƯ VIỆN

MƯỢN SÁCH

Phiếu Mượn

Mượn Trả Sách

Mã Phiếu

Mã Sinh Viên

Ghi Chú

Thêm

Sửa

Xoá

Ghi

Hủy bỏ

Giao diện quản lý mượn trả sách

48

TÀI LIỆU THAM KHẢO

<https://viblo.asia/p/uu-diem-nhuoc-diem-cua-stored-procedures-sql-server-L4x5x3yBIBM>

<https://quantrimang.com/hoc/procedure-thu-tuc-trong-sql-server-159768>

<https://quantrimang.com/hoc/function-ham-trong-sql-server-159749>

<https://viblo.asia/p/function-nguoi-dung-tu-dinh-nghia-trong-sql-GrLZDva35k0>

<https://v1study.com/sql-server-bai-hoc-function-ham-tu-tao-trong-sql-server.html>

<https://topdev.vn/blog/trigger-trong-sql/>

[https://aptech.fpt.edu.vn/trigger-la-gi.html#Trigger trong SQL duoc su dung nhu the nao](https://aptech.fpt.edu.vn/trigger-la-gi.html#Trigger%20trong%20SQL%20duoc%20su%20dung%20nhu%20the%20nao)

<https://viblo.asia/p/su-dung-trigger-trong-sql-qua-vi-du-co-ban-aWj538APK6m>