**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**VIỆN ĐÀO TẠO QUỐC TẾ ( NIIE)**

**\*\*\*\*\*\*\*o0o\*\*\*\*\*\*\***

**BÁO CÁO BÀI TẬP**

**MÔN: LẬP TRÌNH NÂNG CAO  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
ĐỀ TÀI: DỰ ÁN QUẢN LÝ SÁCH**

| *Nhóm* | **02 – 21BITV01** |
| --- | --- |
| *Giảng Viên* | **Ths. Nguyễn Lê Thanh Trúc** |
| *Thành viên nhóm* | **Phan Thanh Hiếu 2100003979**  **Nguyễn Tấn Nhã 2100005929** |



**MỤC LỤC**

[**LỜI MỞ ĐẦU** 1](#_heading=h.gjdgxs)

[**I.**](#_heading=h.30j0zll) **Giới Thiệu Dự Án** 2

[**1.**](#_heading=h.1fob9te) **Mục Đích Của Dự Án** 2

[**2.**](#_heading=h.3znysh7) **Phạm Vi Dự Án** 3

[**II.**](#_heading=h.2et92p0) **Tổng quát về dự án quản lí Sách** 4

[**1.**](#_heading=h.tyjcwt) **Khái Niệm** 4

[**2.**](#_heading=h.3dy6vkm) **Mục Đích** 4

[**3.**](#_heading=h.1t3h5sf) **Chức Năng** 5

[**III.**](#_heading=h.4d34og8) **Cơ Sở Dữ Liệu Của Hệ Thống Nhà Sách** 6

[**1.**](#_heading=h.2s8eyo1) **Các thực thể** 6

[**2.**](#_heading=h.17dp8vu) **Các mối quan hệ giữa các thực thể** 6

[**3.**](#_heading=h.3rdcrjn) **Bảng dữ liệu của các thực thể** 7

[**4.**](#_heading=h.26in1rg) **Thiết Kế SQL Diagram** 9

[**5.**](#_heading=h.lnxbz9) **Entity Relationship Diagram ( Sơ đồ quan hệ )** 10

[**IV.**](#_heading=h.35nkun2) **Xây dựng hệ thống quản lý nhà sách bằng Python và SQL Server** 10

[**1.**](#_heading=h.1ksv4uv) **Mô Tả Bảng Kết Nối Với Sql Server** 10

[**2.**](#_heading=h.44sinio) **Chi Tiết Của Dự Án Quản Lí Nhà Sách Bằng Python Và Sql Server** 11

[**3.**](#_heading=h.2jxsxqh) **Các Chức Năng Menu Của Quản Lí Sách** 17

[**a)**](#_heading=h.z337ya) **Hiển Thị Danh Sách** 17

[**b)**](#_heading=h.3j2qqm3) **Cập Nhật Thêm Sách** 19

[**c)**](#_heading=h.1y810tw) **Tìm Kiếm Thông Tin Sách Theo Mã Sách** 20

[**d)**](#_heading=h.4i7ojhp) **Cập Nhật Đơn Giá Sách Theo Mã Sách** 21

[**e)**](#_heading=h.2xcytpi) **Xóa Một Sản Phẩm Theo Mã Sách** 22

[**f)**](#_heading=h.1ci93xb) **Thoát Chương Trình** 23

[**4.**](#_heading=h.3whwml4) **Tạo Biểu Đồ Thể Hiện Các Thông Tin Trong Cơ Sở Dữ Liệu** 24

[**a)**](#_heading=h.2bn6wsx) **Biểu đồ Cột ( Bar)** 24

[**b)**](#_heading=h.qsh70q) **Biểu đồ Tròn ( Pie )** 26

[**V.**](#_heading=h.1pxezwc) **Kết Luận** 29

# **LỜI MỞ ĐẦU**

Môn học Lập trình nâng cao là một trong những môn học quan trọng và cần thiết trong lĩnh vực Công nghệ thông tin. Đây là môn học cung cấp cho các sinh viên những kiến thức và kỹ năng lập trình phức tạp, giúp cho họ có thể tiếp cận và hiểu rõ hơn về các kỹ thuật lập trình mới nhất và tinh vi hơn. Ngoài việc cung cấp kiến thức và kỹ năng lập trình phức tạp, môn học Lập trình nâng cao còn giúp các sinh viên phát triển tư duy logic và khả năng giải quyết vấn đề, từ đó giúp cho họ trở thành những lập trình viên chuyên nghiệp và có thể đáp ứng được các yêu cầu công việc trong lĩnh vực Công nghệ thông tin. Và với dự án quản lí sách, chúng ta sẽ được trải nghiệm thực tế những kiến thức đó. Đây là một dự án thực sự thú vị, đòi hỏi các sinh viên phải áp dụng những kỹ thuật lập trình nâng cao như lập trình hướng đối tượng, cơ sở dữ liệu, giao diện đồ họa, và nhiều kỹ thuật khác để xây dựng một ứng dụng quản lí sách hoàn chỉnh. Với dự án này, các sinh viên sẽ được hướng dẫn từ cách tạo ra các đối tượng sách và tác giả, đến cách lưu trữ và tìm kiếm các thông tin này trong cơ sở dữ liệu. Ngoài ra, các sinh viên còn được hướng dẫn cách xây dựng giao diện đồ họa cho ứng dụng, để tạo ra trải nghiệm người dùng tốt nhất có thể. Với sự hỗ trợ của giảng viên và đội ngũ trợ giảng giàu kinh nghiệm, sinh viên sẽ có cơ hội học hỏi và trau dồi kỹ năng lập trình một cách toàn diện. Tất cả những kiến thức và kỹ năng này sẽ rất hữu ích cho các sinh viên khi tham gia vào các dự án phát triển phần mềm trong tương lai. môn học Lập trình nâng cao còn đòi hỏi các sinh viên phải có khả năng làm việc độc lập, đồng thời có khả năng hợp tác và làm việc nhóm để xây dựng những ứng dụng phần mềm hoàn chỉnh và hiệu quả. Tóm lại, môn học Lập trình nâng cao và dự án quản lí sách là những môn học quan trọng và thú vị, giúp các sinh viên phát triển kỹ năng lập trình cao cấp, giúp cho họ có thể trở thành những lập trình viên tài năng trong tương lai. Chúc các bạn thành công trong việc học tập và phát triển sự nghiệp!

1. **Giới Thiệu Dự Án**

Dự án quản lí sách là một đề tài rất thú vị và có ý nghĩa vì sách là một tài sản quan trọng và việc quản lý chứng đòi hỏi sự phức tạp và công phu. Qua đó, đề tài này có thể giải quyết một số khó khăn trong việc quản lí sách bằng cách sử dụng công nghệ thông tin và các công cụ quản lí hiện đại để giúp cho quá trình quản lí sách dễ dàng hơn và tiện lợi hơn cho người sử dụng. Bên cạnh đó, đề tài này còn giúp cho người tham gia hiểu rõ hơn về cách thức hoạt động của các công cụ và kỹ thuật quản lí sách hiện đại, cũng như cách áp dụng chúng vào thực tiễn.

1. **Mục Đích Của Dự Án**

Mục đích của Dự án là xây dựng một hệ thống quản lí sách hiệu quả và tiện lợi cho các tổ chức hoặc cá nhân quản lý sách. Hệ thống này sẽ giúp cho người dùng có thể quản lý và sử dụng sách một cách dễ dàng, đồng thời giảm thiểu thời gian và công sức của người quản lý sách.

Các mục tiêu cụ thể của nghiên cứu gồm:

* Phân tích yêu cầu quản lí sách: Tìm hiểu về các yêu cầu cần thiết của người quản lí sách và người sử dụng sách.
* Thiết kế hệ thống quản lí sách: Xây dựng một hệ thống quản lí sách hiệu quả, tiện lợi, đảm bảo tính bảo mật và đáp ứng các yêu cầu của người dùng.
* Phát triển hệ thống: Phát triển hệ thống quản lí sách trên nền tảng công nghệ thông tin hiện đại.
* Đánh giá hiệu quả hệ thống: Đánh giá hiệu quả của hệ thống quản lí sách thông qua việc so sánh với các phương pháp truyền thống và đo lường các chỉ số hiệu quả.

1. **Phạm Vi Dự Án**

Phạm vi dự án sẽ tập trung vào việc thiết kế và phát triển một hệ thống quản lí sách trên nền tảng công nghệ thông tin, bao gồm các tính năng sau:

* Quản lí thông tin sách: Bao gồm các thông tin về tên sách, tác giả, năm xuất bản, số lượng sách, thể loại, giá bán.
* Quản lí thông tin người đọc: Bao gồm các thông tin về tên người đọc, giới tính, ngày sinh.
* Quản lí thông tin mượn/trả sách: Bao gồm các thông tin về người mượn sách, sách mượn/trả, số lượng sách mượn/trả, ngày mượn/trả sách.
* Thống kê báo cáo: Cung cấp các báo cáo thống kê về lượng sách tồn kho, số sách mượn/trả, doanh thu bán sách.
* Tính năng tìm kiếm: Cung cấp tính năng tìm kiếm sách theo các thông tin như tên sách, tên tác giả, năm xuất bản, thể loại.

Với các bảng dữ liệu được tạo ra, hệ thống quản lý sách sẽ có thể cung cấp cho các cửa hàng sách một cách tiếp cận hiệu quả để quản lý thông tin về các sách có sẵn trong kho hàng của họ, quản lý thông tin của người đọc và các giao dịch mượn sách. Các chức năng quản lý sách được cung cấp bao gồm quản lý danh mục thể loại sách, thông tin sách, số lượng sách, năm xuất bản và giá bán. Các thông tin này sẽ giúp cho người quản lý cửa hàng sách có thể kiểm tra tình trạng kho hàng của mình và quản lý chúng một cách hiệu quả. Ngoài ra, hệ thống còn cung cấp chức năng quản lý nhân viên để quản lý thông tin của nhân viên trong cửa hàng sách. Hệ thống còn cho phép quản lý các giao dịch nhập sách vào kho hàng, quản lý thông tin người đọc và các giao dịch mượn sách, bao gồm ngày mượn, ngày trả và số lượng sách mượn. Với các tính năng này, hệ thống quản lý sách sẽ giúp các cửa hàng sách tối ưu hóa quy trình kinh doanh và quản lý thông tin một cách hiệu quả, đồng thời giúp cửa hàng sách nâng cao năng suất và tối đa hóa lợi nhuận.

1. **Tổng quát về dự án quản lý Sách**
2. **Khái Niệm**

* Quản lý nhà sách bằng Python và SQL Server là một hệ thống phần mềm được sử dụng để quản lý thông tin về sách, khách hàng, nhân viên và quản lý kho hàng. Hệ thống này được xây dựng bằng ngôn ngữ lập trình Python và sử dụng cơ sở dữ liệu SQL Server để lưu trữ dữ liệu.
* Với hệ thống này, người dùng có thể thêm, sửa và xóa thông tin về sách, khách hàng, nhân viên và quản lý kho hàng. Hệ thống cũng cho phép người dùng tạo và lập hóa đơn bán hàng, thống kê doanh thu và lợi nhuận của cửa hàng sách.
* Hệ thống quản lý nhà sách bằng Python và SQL Server cung cấp nhiều tính năng hữu ích như tìm kiếm sách theo tên, tác giả hoặc thể loại, xem thông tin chi tiết của từng cuốn sách và kiểm tra số lượng tồn kho. Ngoài ra, hệ thống cũng hỗ trợ quản lý phiếu mượn sách và theo dõi lịch sử mượn trả sách của khách hàng.
* Với sự tích hợp của ngôn ngữ lập trình Python và cơ sở dữ liệu SQL Server, hệ thống quản lý nhà sách đảm bảo tính bảo mật và độ ổn định của dữ liệu. Hơn nữa, việc sử dụng Python cho phần mềm cũng mang lại sự linh hoạt trong việc phát triển và bảo trì hệ thống.

1. **Mục Đích**

* Mục đích của hệ thống quản lý nhà sách bằng Python và SQL Server là cung cấp một giải pháp hiệu quả cho việc quản lý thông tin của một cửa hàng sách. Hệ thống này giúp cửa hàng có thể quản lý thông tin về sách, khách hàng, nhân viên và quản lý kho hàng một cách dễ dàng và chính xác.
* Mục đích chính của hệ thống là giúp cửa hàng sách tăng hiệu quả trong việc quản lý và vận hành hoạt động kinh doanh. Hệ thống cho phép người dùng nhanh chóng tìm kiếm thông tin về sách, quản lý kho hàng và lập hóa đơn bán hàng. Ngoài ra, hệ thống còn cung cấp các chức năng quản lý phiếu mượn sách, thống kê doanh thu và lợi nhuận, giúp cho cửa hàng có thể nắm bắt tình hình kinh doanh và đưa ra các quyết định phù hợp.
* Sử dụng Python và SQL Server cho hệ thống quản lý nhà sách cũng mang lại nhiều lợi ích như tính linh hoạt, khả năng mở rộng và bảo mật dữ liệu cao. Vì vậy, mục đích của hệ thống là cung cấp một giải pháp quản lý thông tin hiệu quả cho cửa hàng sách và giúp tăng cường hoạt động kinh doanh của cửa hàng.

1. **Chức Năng**

* Hệ thống quản lý nhà sách bằng Python và SQL Server có những chức năng sau:
* *Quản lý thông tin sách*: cho phép người dùng thêm, xóa, sửa thông tin sách, bao gồm mã sách, tên sách, thể loại, tác giả, năm xuất bản, giá bán, số lượng tồn kho và các thông tin khác liên quan đến sách.
* *Quản lý thông tin người đọc*: cho phép người dùng thêm, xóa, sửa thông tin người đọc, bao gồm mã khách, tên, giới tính, ngày sinh, số điện thoại và các thông tin khác liên quan đến người đọc.
* *Quản lý thông tin nhân viên*: cho phép người dùng thêm, xóa, sửa thông tin nhân viên, bao gồm mã nhân viên, tên, số điện thoại, ngày sinh và các thông tin khác liên quan đến nhân viên.
* *Quản lý phiếu mượn sách*: cho phép người dùng lập phiếu mượn sách, bao gồm thông tin về người mượn, sách mượn, số lượng, ngày mượn, ngày trả và mã nhân viên thực hiện. Hệ thống cũng cho phép người dùng tìm kiếm thông tin phiếu mượn sách và thực hiện trả sách.
* *Các chức năng khác*: hệ thống còn cung cấp các chức năng khác như tìm kiếm thông tin sách, thống kê số lượng sách, doanh thu và lợi nhuận, cập nhật thông tin kho sách, báo cáo tồn kho và các chức năng quản lý khác để hỗ trợ cho hoạt động kinh doanh của cửa hàng sách.

1. **Cơ Sở Dữ Liệu Của Hệ Thống Nhà Sách**
2. **Các thực thể**

* Thực thể (thể loại sách): bao gồm mã thể loại và tên thể loại, thể hiện thông tin về các thể loại sách hiện có trong cửa hàng sách.
* Thực thể (sách): bao gồm đầy đủ các thông tin về từng quyển sách như mã sách, tên sách, thể loại, tác giả, năm xuất bản, số lượng trong kho và giá tiền của từng quyển sách.
* Thực thể (nhân viên): bao gồm các thông tin về nhân viên của cửa hàng sách như mã nhân viên, tên nhân viên, ngày sinh và số điện thoại.
* Thực thể (nhập sách): được sử dụng để quản lý thông tin khi nhập sách vào kho, bao gồm mã sách, tên sách, ngày nhập và số lượng.
* Thực thể (người đọc): bao gồm các thông tin về người đọc như mã người đọc, tên, giới tính và ngày sinh.
* Thực thể (phiếu mượn sách): bao gồm các thông tin về phiếu mượn sách như mã phiếu, mã sách, mã người đọc, tên người đọc, ngày mượn, ngày trả, số lượng sách được mượn và mã nhân viên thực hiện giao dịch.

1. **Các mối quan hệ giữa các thực thể**

* Quan hệ (nhiều-nhiều) giữa thể loại và sách vì 1 tựa sách có thể có nhiều thể loại và 1 thể loại có thể có nhiều tựa sách khác nhau
* Quan hệ (một-nhiều) giữa nhập sách và sách vì một lần nhập sách có thể nhập nhiều sách.
* Quan hệ (một-một) giữa phiếu mượn sách và người đọc vì mỗi người đọc chỉ được mượn một lần và trả sách sau đó mới được tiếp tục mượn.
* Quan hệ (một-nhiều) giữa nhân viên và phiếu mượn sách vì mỗi nhân viên có thể duyệt nhiều yêu cầu mượn sách khác nhau.
* Quan hệ (nhiều-nhiều) giữa phiếu mượn sách và sách vì mỗi phiếu mượn sách có thể mượn nhiều cuốn sách và mỗi quyển sách có thể được mượn bởi nhiều người qua phiếu mượn sách.

1. **Bảng dữ liệu của các thực thể**

* **Bảng sách:**

| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã sách | Int, primary key |
| Tên sách | Nvarchar(100) |
| Giá | Int |
| Năm xuất bản | Int |
| Số lượng | Int |
| Thể loại | Int, foreign key |

* **Bảng thể loại:**

| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã thể loại | Int, primary key |
| Tên thể loại | Nvarchar(100) |

* **Bảng nhân viên:**

| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã nhân viên | Int, primary |
| Họ tên | Nvarchar(100) |
| Ngày sinh | Date |
| Số điện thoại | Varchar(20) |

* **Bảng nhập sách:**

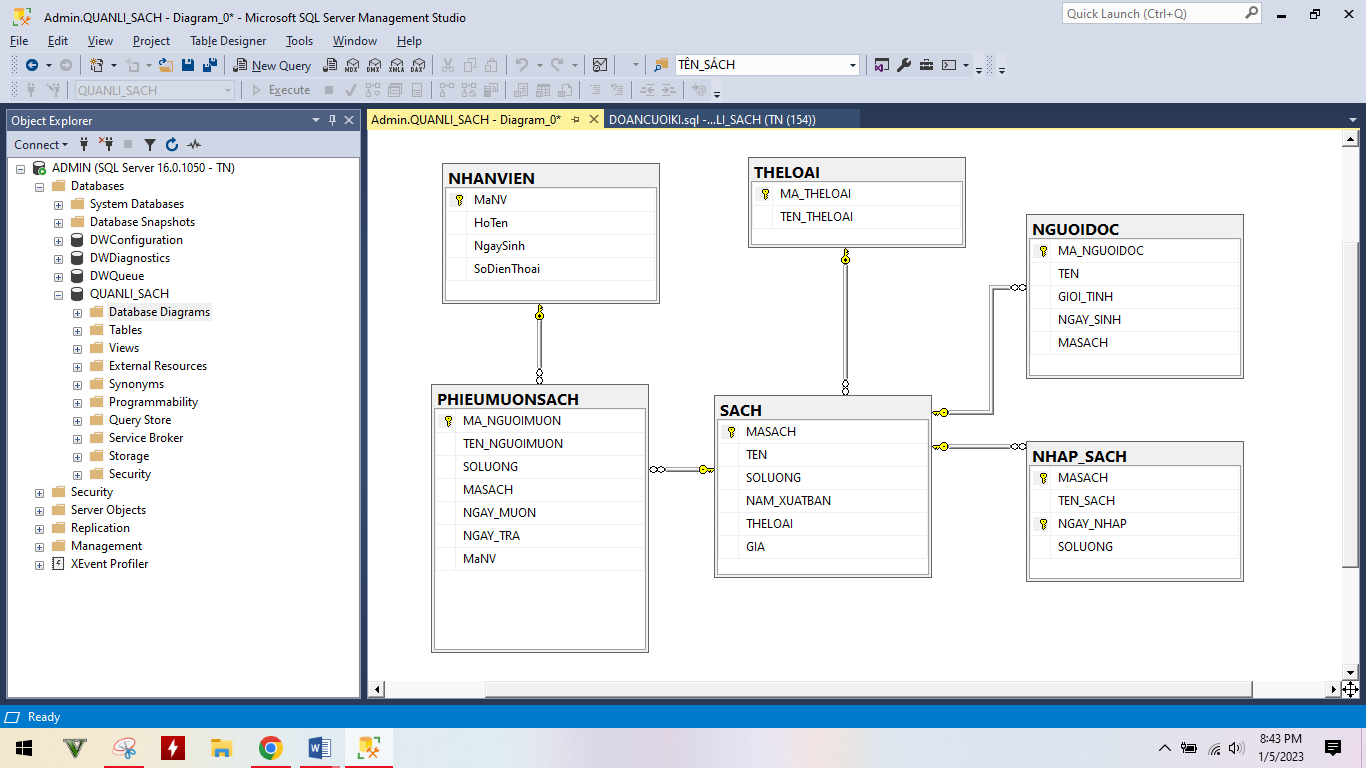
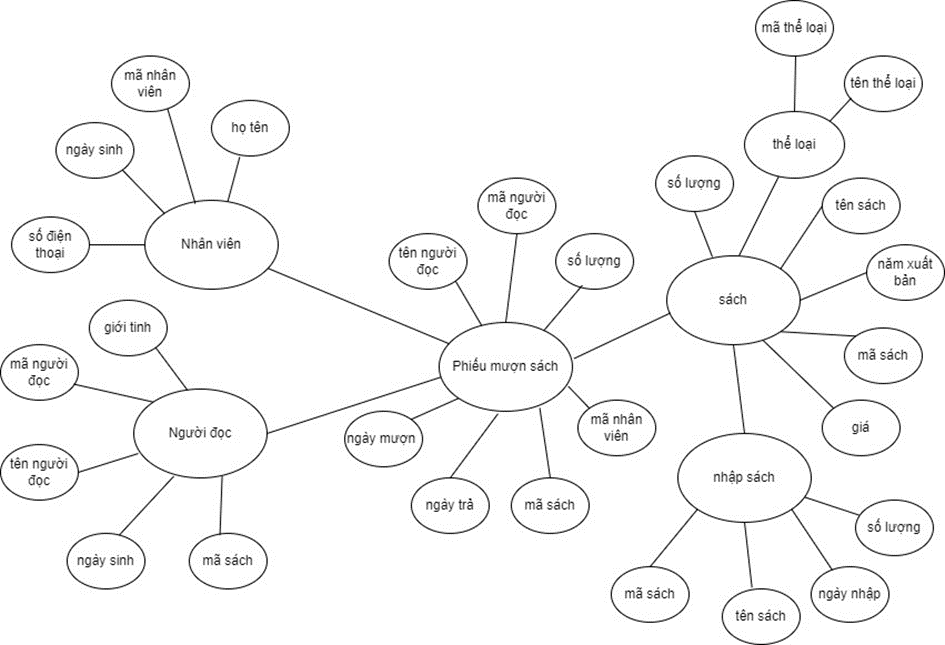
| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã sách | Int, primary key, foreign key |
| Tên sách | nvarchar(100) |
| Ngày nhập | Date, primary key |
| Số lượng | Int |

* **Bảng người đọc:**

| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã người đọc | Int primary key |
| Tên | Nvarchar(100) |
| Giới tính | Int |
| Ngày sinh | Date |
| Mã sách | Int, foreign key |

* **Bảng phiếu mượn sách:**

| Thực thể | Kiểu dữ liệu |
| --- | --- |
| Mã người đọc | Int primary key |
| Tên người đọc | Nvarchar(100) |
| Số lượng | Int |
| Ngày mượn | Date |
| Ngày trả | Date |
| Mã sách | Int, foreign key |
| Mã nhân viên | Int, foreign key |

1. **Thiết Kế SQL Diagram**
2. **Entity Relationship Diagram ( Sơ đồ quan hệ )**

1. **Xây dựng hệ thống quản lý nhà sách bằng Python và SQL Server**
2. **Mô Tả Bảng Kết Nối Với Sql Server**

Xây dựng hệ thống quản lý nhà sách sử dụng Python kết hợp với SQL Server để lưu trữ dữ liệu. Hệ thống này sẽ bao gồm các chức năng quản lý thông tin sách, thông tin người đọc, thông tin nhân viên và phiếu mượn sách.

Đây là một ví dụ về cách sử dụng module pyodbc trong Python để tạo và kết nối với cơ sở dữ liệu SQL Server. Ví dụ này tạo ra 5 bảng: "SACH", "NHANVIEN", "NHAP\_SACH", "NGUOIDOC" và "PHIEUMUONSACH". Mỗi bảng được định nghĩa bởi một tập các cột và ràng buộc khóa ngoại.

* Bảng THELOAI bao gồm hai cột: MA\_THELOAI và TEN\_THELOAI. Cột MA\_THELOAI có kiểu dữ liệu INTEGER và là khóa chính của bảng.
* Bảng SACH gồm 6 cột: MASACH (khóa chính, kiểu INTEGER), TEN (tên sách, kiểu NVARCHAR(100), bắt buộc), SOLUONG (số lượng sách, kiểu INT), NAM\_XUATBAN (năm xuất bản, kiểu INTEGER), THELOAI (mã thể loại, kiểu INTEGER, khóa ngoại tham chiếu đến bảng THELOAI), và GIA (giá tiền, kiểu INTEGER, bắt buộc).
* Bảng NHANVIEN gồm 4 cột: MaNV (khóa chính, kiểu INT), HoTen (họ tên nhân viên, kiểu NVARCHAR(100), bắt buộc), NgaySinh (ngày sinh, kiểu DATE, bắt buộc), và SoDienThoai (số điện thoại, kiểu VARCHAR(20), bắt buộc).
* Bảng NHAP\_SACH bao gồm 4 cột: MASACH (khóa ngoại tham chiếu đến cột MASACH trong bảng SACH), TEN\_SACH (tên sách, kiểu NVARCHAR(100), bắt buộc), NGAY\_NHAP (ngày nhập sách, kiểu DATE, khóa chính cùng với MASACH), và SOLUONG (số lượng sách nhập vào, kiểu INTEGER).
* Bảng NGUOIDOC gồm 5 cột: MA\_NGUOIDOC (khóa chính, kiểu INTEGER), TEN (tên người đọc, kiểu NVARCHAR(100), bắt buộc), GIOI\_TINH (giới tính, kiểu INT), NGAY\_SINH (ngày sinh, kiểu DATE), và MASACH (khóa ngoại tham chiếu đến cột MASACH trong bảng SACH).
* Bảng PHIEUMUONSACH gồm 7 cột: MA\_NGUOIMUON (khóa chính, kiểu INTEGER), TEN\_NGUOIMUON (tên người mượn, kiểu NVARCHAR(100), bắt buộc), SOLUONG (số lượng sách mượn, kiểu INT), MASACH (khóa ngoại tham chiếu đến cột MASACH trong bảng SACH), NGAY\_MUON (ngày mượn sách, kiểu DATE), NGAY\_TRA (ngày trả sách, kiểu DATE), và MaNV (khóa ngoại tham chiếu đến cột MaNV trong bảng NHANVIEN).

1. **Chi Tiết Của Dự Án Quản Lí Nhà Sách Bằng Python Và Sql Server**

* *Tạo* ***class THELOAI*** *được định nghĩa với các thuộc tính và phương thức như sau:*

**Các thuộc tính của lớp THELOAI:**

* MA\_THELOAI: mã thể loại sách
* TEN\_THELOAI: tên thể loại sách
* connection: đối tượng kết nối cơ sở dữ liệu SQL Server
* mycursor: đối tượng con trỏ truy vấn cơ sở dữ liệu
* server: tên server SQL Server
* database: tên cơ sở dữ liệu
* username: tên đăng nhập SQL Server
* password: mật khẩu đăng nhập SQL Server

**Các phương thức của lớp THELOAI:**

* init(self, MA\_THELOAI, TEN\_THELOAI, server, database, username, password): phương thức khởi tạo đối tượng của class THELOAI, được gọi đến khi tạo mới một đối tượng THELOAI. Phương thức này có các tham số truyền vào để thiết lập kết nối đến cơ sở dữ liệu SQL Server và khởi tạo các thuộc tính MA\_THELOAI và TEN\_THELOAI.
* display\_table(self): phương thức thực hiện truy vấn cơ sở dữ liệu để lấy thông tin các thể loại sách từ bảng THELOAI, sau đó sử dụng thư viện prettytable để hiển thị kết quả truy vấn ra màn hình dưới dạng bảng.

**Cách sử dụng đối tượng THELOAI:**

* Tạo đối tượng THELOAI: theloai = THELOAI(MA\_THELOAI, TEN\_THELOAI, server, database, username, password)
* Hiển thị bảng thông tin thể loại sách: theloai.display\_table()
* *Tạo* ***CLASS SACH*** *định nghĩa một đối tượng Sách, bao gồm các thuộc tính mã sách, tên sách, số lượng, năm xuất bản, mã thể loại và giá. Đối tượng Sách có các phương thức như sau:*
* Phương thức khởi tạo Init: khởi tạo đối tượng Sach với các thuộc tính mã sách, tên sách, số lượng, năm xuất bản, mã thể loại và giá.
* Phương thức them\_sach: thêm thông tin sách mới vào cơ sở dữ liệu bằng câu lệnh SQL INSERT INTO.
* Phương thức xoa\_sach: xóa thông tin sách khỏi cơ sở dữ liệu bằng câu lệnh SQL DELETE FROM.
* Phương thức hien\_thi\_sach: hiển thị thông tin của tất cả các sách có trong cơ sở dữ liệu bằng câu lệnh SQL SELECT \* FROM.
* Phương thức tim\_sach\_theo\_ma: tìm kiếm thông tin của một sách dựa vào mã sách nhập từ bàn phím bằng câu lệnh SQL SELECT \* FROM WHERE MASACH =
* Phương thức cap\_nhat\_gia\_sach: cập nhật giá mới cho sách dựa vào mã sách và giá mới nhập từ bàn phím bằng câu lệnh SQL UPDATE SET GIA = WHERE MASACH =.

**Các thuộc tính của lớp SACH bao gồm:**

* ma\_sach: mã sách
* ten\_sach: tên sách
* so\_luong: số lượng sách
* nam\_xuat\_ban: năm xuất bản
* ma\_theloai: mã thể loại của sách
* gia: giá tiền của sách

**Các phương thức của lớp SACH bao gồm:**

* them\_sach(): thêm thông tin sách vào cơ sở dữ liệu
* xoa\_sach(): xóa thông tin sách khỏi cơ sở dữ liệu dựa trên mã sách
* hien\_thi\_sach(): hiển thị tất cả thông tin sách trong cơ sở dữ liệu
* tim\_sach\_theo\_ma(): tìm kiếm sách dựa trên mã sách
* cap\_nhat\_gia\_sach(): cập nhật giá tiền của sách

**Cách sử dụng đối tượng SACH:**

* Khởi tạo đối tượng SACH với các thông tin về sách
* Sử dụng các phương thức của đối tượng SACH để thực hiện các thao tác trên cơ sở dữ liệu như thêm, xóa, hiển thị, tìm kiếm, cập nhật thông tin sách.
* *Class NHANVIEN là một lớp trong Python đại diện cho bảng nhân viên trong cơ sở dữ liệu. Lớp này bao gồm các thuộc tính và phương thức để thao tác với dữ liệu trong bảng nhân viên.*

**Các thuộc tính của lớp NHANVIEN gồm:**

* MaNV: mã nhân viên
* HoTen: họ tên nhân viên
* NgaySinh: ngày sinh của nhân viên
* SoDienThoai: số điện thoại của nhân viên

**Các phương thức của lớp NHANVIEN bao gồm:**

* init(self, MaNV, HoTen, NgaySinh, SoDienThoai): phương thức khởi tạo, nhận vào các tham số MaNV, HoTen, NgaySinh, SoDienThoai để khởi tạo các thuộc tính tương ứng của đối tượng NHANVIEN.
* them\_nhanvien(self): phương thức để thêm thông tin của nhân viên mới vào cơ sở dữ liệu.
* sua\_nhanvien(self, field, new\_value): phương thức để cập nhật thông tin của một nhân viên trong cơ sở dữ liệu, nhận vào tham số field để chỉ ra thông tin cần cập nhật và new\_value để chỉ ra giá trị mới.
* xoa\_nhanvien(self): phương thức để xóa thông tin của một nhân viên khỏi cơ sở dữ liệu.
* hien\_thi\_nhanvien(self): phương thức để hiển thị tất cả thông tin của các nhân viên có trong cơ sở dữ liệu.

**Tham số được truyền vào phương thức khởi tạo:**

* MaNV: Mã số nhân viên (kiểu dữ liệu: str)
* HoTen: Họ và tên của nhân viên (kiểu dữ liệu: str)
* NgaySinh: Ngày sinh của nhân viên (kiểu dữ liệu: str)
* SoDienThoai: Số điện thoại của nhân viên (kiểu dữ liệu: str)

Để tạo một đối tượng NHANVIEN, ta cần khởi tạo một instance (thể hiện) của class NHANVIEN bằng cách truyền các tham số vào phương thức khởi tạo của lớp này. Sau đó, ta có thể gọi các phương thức của đối tượng để thực hiện các chức năng như thêm, sửa, xóa, hiển thị thông tin của nhân viên.

* *Tạo Class NHAP\_SACH định nghĩa một đối tượng để quản lý thông tin nhập sách vào kho.*

**Các thuộc tính của lớp NHAP\_SACH gồm:**

* MASACH: mã sách
* TEN\_SACH: tên sách
* NGAY\_NHAP: ngày nhập sách
* SOLUONG: số lượng sách nhập vào kho

**Các phương thức của lớp NHAP\_SACH gồm:**

* init(self, MASACH, TEN\_SACH, NGAY\_NHAP, SOLUONG): phương thức khởi tạo đối tượng với các thông số đầu vào là mã sách, tên sách, ngày nhập và số lượng.
* nhap\_sach\_moi(self): phương thức để thêm thông tin của một cuốn sách mới vào kho. Phương thức sẽ yêu cầu người dùng nhập các thông tin về sách mới như mã sách, tên sách, số lượng và giá tiền. Sau đó, phương thức thêm thông tin sách mới vào cơ sở dữ liệu.
* hien\_thi\_sach(self): phương thức này hiển thị tất cả thông tin của các cuốn sách trong kho.
* *Tạo Class NGUOIDOC định nghĩa một đối tượng người đọc trong hệ thống thư viện. Đối tượng này chứa thông tin về mã người đọc, tên, giới tính, ngày sinh và mã sách đang mượn.*

**Các thuộc tính của lớp NGUOIDOC:**

* ma\_nguoidoc (str): Mã người đọc.
* ten (str): Tên người đọc.
* gioi\_tinh (str): Giới tính của người đọc.
* ngay\_sinh (str): Ngày sinh của người đọc.
* masach (str): Mã sách đang mượn.

**Các phương thức NGUOIDOC bao gồm:**

* init(self, ma\_nguoidoc, ten, gioi\_tinh, ngay\_sinh, masach): phương thức khởi tạo đối tượng, nhận vào các tham số để khởi tạo giá trị cho các thuộc tính tương ứng. Ngoài ra, phương thức này còn thiết lập kết nối tới cơ sở dữ liệu.
* insert(self): phương thức thêm một bản ghi mới vào cơ sở dữ liệu với các giá trị tương ứng của các thuộc tính. Phương thức này sử dụng câu lệnh SQL INSERT để thêm bản ghi mới.
* update(self): phương thức cập nhật một bản ghi trong cơ sở dữ liệu với các giá trị mới của các thuộc tính. Phương thức này sử dụng câu lệnh SQL UPDATE để cập nhật bản ghi.
* delete(self): phương thức xóa một bản ghi khỏi cơ sở dữ liệu với mã người đọc tương ứng. Phương thức này sử dụng câu lệnh SQL DELETE để xóa bản ghi.
* get\_all(): phương thức trả về tất cả các bản ghi trong bảng NGUOIDOC dưới dạng một danh sách các đối tượng NGUOIDOC. Phương thức này sử dụng câu lệnh SQL SELECT để lấy tất cả các bản ghi và sau đó tạo ra danh sách các đối tượng NGUOIDOC tương ứng.

Để sử dụng đối tượng NGUOIDOC, ta khởi tạo một đối tượng mới với các thông tin tương ứng và sử dụng các phương thức để thêm, cập nhật hoặc xóa thông tin của đối tượng đó khỏi cơ sở dữ liệu.

* Lớp *PHIEUMUONSACH* đại diện cho bảng thông tin phiếu mượn sách trong hệ thống thư viện.

**Lớp PHIEUMUONSACH có các thuộc tính:**

* MA\_NGUOIMUON: mã người mượn sách
* TEN\_NGUOIMUON: tên người mượn sách
* SOLUONG: số lượng sách mượn
* MASACH: mã sách mượn
* NGAY\_MUON: ngày mượn sách
* NGAY\_TRA: ngày trả sách
* MaNV: mã nhân viên thực hiện việc mượn sách

**Các phương thức của lớp PHIEUMUONSACH bao gồm:**

* init(self, MA\_NGUOIMUON, TEN\_NGUOIMUON, SOLUONG, MASACH, NGAY\_MUON, NGAY\_TRA, MaNV): phương thức khởi tạo đối tượng PHIEUMUONSACH với các thuộc tính tương ứng.
* save(self): phương thức lưu thông tin phiếu mượn sách vào cơ sở dữ liệu.
* get\_all(): phương thức lấy tất cả các bản ghi trong bảng PHIEUMUONSACH từ cơ sở dữ liệu và trả về một danh sách các đối tượng PHIEUMUONSACH.

1. **Các Chức Năng Menu Của Quản Lí Sách**
2. **Hiển Thị Danh Sách**

Chức năng "hiển thị danh sách sách" có nhiệm vụ lấy dữ liệu từ cơ sở dữ liệu về tất cả các cuốn sách và hiển thị thông tin của chúng lên một bảng đẹp mắt để người dùng dễ dàng quan sát và tra cứu. giúp người dùng có thể xem thông tin về tất cả các cuốn sách đang có trong thư viện, bao gồm mã sách, tên sách, số lượng, năm xuất bản, thể loại và giá.

Đoạn code trên trình bày chức năng này bằng cách sử dụng SQL để lấy dữ liệu từ bảng SACH trong cơ sở dữ liệu. Sau đó, nó sử dụng thư viện prettytable để tạo ra một bảng đẹp mắt để hiển thị thông tin cho người dùng.

Cụ thể, hàm "hien\_thi\_sach" của lớp Sach thực hiện các bước sau:

* Tạo một truy vấn SQL để lấy toàn bộ dữ liệu từ bảng SACH. Tạo một chuỗi SQL để truy vấn tất cả các dữ liệu trong bảng "SACH".
* Thực thi truy vấn và lấy tất cả các dòng dữ liệu từ kết quả truy vấn. Sử dụng phương thức execute() để thực thi câu lệnh SQL.
* Sử dụng phương thức fetchall() để lấy tất cả các kết quả của câu lệnh truy vấn và lưu trữ chúng trong biến result.
* Sử dụng thư viện prettytable để tạo một bảng mới có các tiêu đề tương ứng với các cột của bảng SACH.
* Với mỗi dòng dữ liệu lấy được từ kết quả truy vấn, thêm một hàng mới vào bảng và điền các giá trị tương ứng vào các ô của hàng đó.
* Hiển thị bảng đó ra màn hình để người dùng có thể xem thông tin các cuốn sách có trong cơ sở dữ liệu.
* Cuối cùng, đóng kết nối và con trỏ chuột của đối tượng cơ sở dữ liệu.

Với các định dạng chuỗi {:<15}, :<20, ... được sử dụng để canh lề cho phù hợp với độ dài của mỗi giá trị. Cụ thể, độ dài của mỗi cột được chỉ định để tạo bảng được đẹp hơn. Chức năng hiển thị thông tin sách từ một bảng cơ sở dữ liệu:

* *def hien\_thi\_sach(self):* Khai báo một method (phương thức) với tên là "hien\_thi\_sach" trong một class (lớp). Method này được sử dụng để hiển thị thông tin sách.
* *sql = "SELECT \* FROM SACH"*: Khai báo một chuỗi chứa câu lệnh SQL để truy vấn toàn bộ dữ liệu từ bảng "SACH" trong cơ sở dữ liệu.
* *self.mycursor.execute(sql):* Thực thi câu lệnh SQL bằng cách gọi phương thức "execute" của đối tượng "mycursor". Đối tượng này đại diện cho một con trỏ trỏ đến kết nối đến cơ sở dữ liệu.
* *rows = self.mycursor.fetchall():* Lấy tất cả các dòng dữ liệu kết quả của câu lệnh truy vấn bằng cách gọi phương thức "fetchall" của đối tượng "mycursor".
* *table = prettytable.PrettyTable(["MASACH", "TEN", "SOLUONG", "NAM\_XUATBAN", "THELOAI", "GIA"]*): Tạo một đối tượng PrettyTable để lưu trữ và hiển thị dữ liệu sách. Đối tượng này có các cột được định nghĩa bởi danh sách các tên cột, bao gồm "MASACH", "TEN", "SOLUONG", "NAM\_XUATBAN", "THELOAI" và "GIA".
* *for row in rows: table.add\_row(row):* Với mỗi dòng dữ liệu được lấy từ kết quả truy vấn, thêm một hàng mới vào PrettyTable bằng cách gọi phương thức "add\_row" của đối tượng PrettyTable.
* *print(table):* In ra bảng dữ liệu sách được định dạng sử dụng đối tượng PrettyTable.

1. **Cập Nhật Thêm Sách**

Chức năng "Cập nhật thêm sách" được sử dụng để thêm mới thông tin của một cuốn sách vào cơ sở dữ liệu sách đã có. Cụ thể, chức năng này có thể bao gồm các bước sau:

* Yêu cầu người dùng nhập thông tin của cuốn sách cần thêm vào cơ sở dữ liệu. Thông tin này có thể bao gồm mã sách, tên sách, số lượng, năm xuất bản, thể loại, giá, v.v.
* Kiểm tra xem cuốn sách có tồn tại trong cơ sở dữ liệu hay chưa. Nếu cuốn sách đã tồn tại, có thể cập nhật lại thông tin của nó. Nếu chưa tồn tại, thêm thông tin của cuốn sách mới vào cơ sở dữ liệu.
* Sau khi kiểm tra và cập nhật/thêm mới thành công, thông báo cho người dùng biết rằng việc cập nhật/thêm mới đã hoàn tất.
* Nếu trong quá trình cập nhật/thêm mới có lỗi xảy ra, hiển thị thông báo lỗi cho người dùng và hướng dẫn người dùng khắc phục sự cố.

Chức năng "Cập nhật thêm sách" giúp cập nhật và bảo trì thông tin của cơ sở dữ liệu sách, đảm bảo tính an toàn và chính xác của thông tin, từ đó hỗ trợ quản lý và tìm kiếm thông tin sách một cách dễ dàng và thuận tiện hơn. Ngoài ra, việc thêm mới sách vào cơ sở dữ liệu cũng giúp mở rộng thư viện sách, đáp ứng nhu cầu đa dạng của người dùng. Chức năng của đoạn code trên là thêm một quyển sách mới vào cơ sở dữ liệu:

* *def them\_sach(self)::* đây là hàm để thêm sách mới vào cơ sở dữ liệu.
* *sql = "INSERT INTO SACH (MASACH, TEN, SOLUONG, NAM\_XUATBAN, THELOAI, GIA) VALUES (?, ?, ?, ?, ?, ?)":* đây là câu lệnh SQL để thêm một quyển sách mới vào bảng SACH trong cơ sở dữ liệu. Các giá trị sẽ được đưa vào bằng cách sử dụng các tham số định dạng.
* *values = (self.ma\_sach, self.ten\_sach, self.so\_luong, self.nam\_xuat\_ban, self.ma\_theloai, self.gia)*: đây là các giá trị được truyền vào trong câu lệnh SQL. Giá trị của mỗi tham số được lấy từ các thuộc tính tương ứng của đối tượng sách.
* *self.mycursor.execute(sql, values):* đây là phương thức execute của đối tượng cursor để thực thi câu lệnh SQL với các giá trị được truyền vào.
* *self.connection.commit():* đây là phương thức commit của đối tượng connection để xác nhận các thay đổi được thực hiện trong cơ sở dữ liệu.
* *print(f"Thêm sách {self.ten\_sach} thành công!"):* đây là thông báo được hiển thị trên màn hình để cho biết sách đã được thêm thành công vào cơ sở dữ liệu.

1. **Tìm Kiếm Thông Tin Sách Theo Mã Sách**

Chức năng “tìm kiếm thông tin sách theo mã sách” nhằm giúp người dùng có thể tìm kiếm thông tin về một cuốn sách trong cơ sở dữ liệu dựa trên mã sách được cung cấp. Các bước chính để thực hiện chức năng này bao gồm:

* Người dùng nhập mã sách cần tìm kiếm thông tin vào một trình nhập liệu, ví dụ như một ô nhập liệu trên giao diện người dùng.
* Mã sách được chuyển đến phương thức tìm kiếm trong cơ sở dữ liệu.
* Phương thức tìm kiếm truy vấn cơ sở dữ liệu để tìm kiếm tất cả các thông tin liên quan đến cuốn sách có mã sách tương ứng.
* Kết quả tìm kiếm được trả về và hiển thị cho người dùng, ví dụ như trên giao diện người dùng hoặc thông qua một bản in.
* Việc tìm kiếm thông tin sách theo mã sách được giả định là đã được cài đặt trong chương trình, vì vậy nó sẽ liên quan đến các lệnh truy vấn cơ sở dữ liệu để tìm kiếm sách có mã sách tương ứng. Sau khi tìm kiếm được thông tin sách theo mã sách, thông tin đó có thể được trả về cho người dùng hoặc được sử dụng trong các chức năng khác của chương trình, ví dụ như để hiển thị thông tin chi tiết về một cuốn sách trên giao diện người dùng hoặc để cho phép người dùng mượn sách. Hàm tìm\_sach\_theo\_ma có chức năng tìm kiếm thông tin của một quyển sách trong cơ sở dữ liệu dựa trên mã sách:
* Dòng *ma\_sach = input("Nhập mã sách: ")* yêu cầu người dùng nhập mã sách cần tìm kiếm.
* Dòng *sql = "SELECT \* FROM SACH WHERE MASACH = ?"* khai báo câu lệnh SQL để truy vấn thông tin sách trong cơ sở dữ liệu với điều kiện mã sách trùng với mã sách vừa nhập.
* Dòng *values = (ma\_sach, )* khai báo tuple values để truyền vào câu lệnh SQL, với giá trị của phần tử là ma\_sach.
* Dòng *self.mycursor.execute(sql, values)* thực thi câu lệnh SQL với giá trị của values.
* Dòng *result = self.mycursor.fetchall()* lấy tất cả các bản ghi được trả về từ câu lệnh SELECT.
* Dòng *if len(result) == 0:* kiểm tra xem có bản ghi nào được tìm thấy không. Nếu không tìm thấy thì in ra thông báo.
* Nếu tìm thấy, hàm sẽ in ra tiêu đề của bảng thông tin sách và dùng vòng lặp for để lấy các giá trị từ các bản ghi trả về và in chúng ra theo định dạng bảng.

1. **Cập Nhật Đơn Giá Sách Theo Mã Sách**

Chức năng Cập Nhật Đơn Giá Sách Theo Mã Sách nhằm cho phép người dùng cập nhật giá của một quyển sách đã có trong hệ thống theo mã sách tương ứng.

Đoạn code thực hiện chức năng này như sau:

* Người dùng nhập mã sách cần cập nhật đơn giá.
* Thực hiện truy vấn cơ sở dữ liệu để lấy thông tin của quyển sách đó dựa trên mã sách vừa nhập.
* Nếu không tìm thấy thông tin sách, thông báo cho người dùng biết.
* Nếu tìm thấy thông tin sách, cho phép người dùng cập nhật đơn giá mới.
* Thực hiện cập nhật đơn giá mới vào cơ sở dữ liệu và thông báo thành công cho người dùng. Nếu cập nhật thất bại, thông báo lỗi tương ứng.

Hàm cap\_nhat\_gia\_sach có chức năng cập nhật giá mới cho một cuốn sách trong cơ sở dữ liệu bằng cách nhập mã sách và giá mới từ bàn phím. Cụ thể:

* Dòng *gia\_moi = input*("Nhập giá mới cho sách: ") yêu cầu người dùng nhập giá mới cho cuốn sách cần cập nhật.
* Dòng *sql = "UPDATE SACH SET GIA = ? WHERE MASACH = ?"* định nghĩa truy vấn SQL để cập nhật giá mới cho cuốn sách trong cơ sở dữ liệu. Trong truy vấn này, dấu ? được sử dụng để đại diện cho giá trị giá mới và mã sách.
* Dòng *values = (gia\_moi, self.ma\_sach)* định nghĩa các giá trị cần truyền vào truy vấn SQL ở trên. Biến gia\_moi là giá mới vừa nhập từ bàn phím, còn self.ma\_sach là mã sách của cuốn sách cần cập nhật giá mới (biến này được khai báo trước đó trong class và có thể được truyền vào hàm từ bên ngoài).
* Dòng *self.mycursor.execute(sql, values)* thực thi truy vấn SQL với các giá trị đã được định nghĩa ở trên.
* Dòng *self.connection.commit()* lưu lại các thay đổi vào cơ sở dữ liệu.
* Dòng *print(f"Đã cập nhật giá mới {gia\_moi} cho sách có mã {self.ma\_sach}!")* thông báo cho người dùng biết rằng giá mới đã được cập nhật thành công cho cuốn sách với mã sách tương ứng.

1. **Xóa Một Sản Phẩm Theo Mã Sách**

Chức năng Xóa Một Sản Phẩm Theo Mã Sách trong một hệ thống quản lý sách thường được sử dụng để xóa một cuốn sách cụ thể khỏi danh sách sách trong kho hoặc thư viện. Chức năng này có thể giúp người quản lý giữ cho danh sách sách của mình luôn được cập nhật và chính xác. Trong phần mềm quản lý sách của bạn, chức năng Xóa Một Sản Phẩm Theo Mã Sách có thể được thực hiện thông qua đoạn mã.

Chức năng này sẽ yêu cầu người dùng nhập mã sách của cuốn sách cần xóa. Sau đó, đoạn mã sử dụng câu lệnh SQL "DELETE" để xóa cuốn sách có mã sách trùng khớp với mã sách được nhập vào. Cuối cùng, đoạn mã sẽ thông báo cho người dùng biết rằng sách đã được xóa thành công khỏi hệ thống.

Hàm xoa\_sach có chức năng xóa một cuốn sách khỏi danh sách sách của thư viện dựa trên mã sách của cuốn sách đó.

* Đầu tiên, hàm sẽ yêu cầu người dùng nhập vào mã sách cần xóa thông qua lệnh input("Nhập mã sách cần xóa: ").
* Sau đó, câu lệnh *SQL "DELETE FROM SACH WHERE MASACH = ?"* được gán cho biến sql. Đây là câu lệnh SQL để xóa một cuốn sách khỏi bảng SACH dựa trên mã sách của cuốn sách đó.
* Tiếp theo, giá trị của mã sách được đưa vào tuple values thông qua lệnh *values = (ma\_sach,).* Tuple này sẽ được sử dụng để chèn giá trị vào câu lệnh SQL.
* Sau đó, lệnh *self.mycursor.execute(sql, values)* được sử dụng để thực thi câu lệnh SQL, trong đó sql là câu lệnh SQL đã được chuẩn bị và values là tuple chứa giá trị cần chèn vào câu lệnh SQL.
* Tiếp theo, *self.connection.commit()* được gọi để lưu các thay đổi vào cơ sở dữ liệu.
* Cuối cùng, thông báo xác nhận đã xóa sách thành công được hiển thị trên màn hình thông qua lệnh *print(f"Đã xóa sách có mã {ma\_sach} thành công!")*.

1. **Thoát Chương Trình**

Hàm Thoát Chương Trình được sử dụng để kết thúc hoạt động của chương trình. Khi được gọi, nó sẽ hiển thị thông báo "Đã thoát chương trình!" và sử dụng lệnh sys.exit() để thoát khỏi chương trình. Đây là một cách tiện lợi để kết thúc chương trình một cách chính xác và đúng cách, đồng thời giúp người dùng nhận biết được chương trình đã kết thúc thành công.

1. **Tạo Biểu Đồ Thể Hiện Các Thông Tin Trong Cơ Sở Dữ Liệu**
2. **Biểu đồ Cột ( Bar)**

Đầu tiên ta kết nối tới cơ sở dữ liệu Microsoft SQL Server, lấy thông tin sách và vẽ biểu đồ cột giá sách theo tên sách sử dụng các thư viện matplotlib và numpy. Dưới đây là hướng dẫn chi tiết từng bước để hiểu rõ hơn về đoạn code này:

1. Nhập các thư viện cần thiết:

***import matplotlib.pyplot as plt***

***import numpy as np***

***import pyodbc***

* Thư viện matplotlib được sử dụng để trực quan hóa dữ liệu, trong khi numpy được sử dụng để tính toán số học.
* Thư viện pyodbc là một thư viện Python để kết nối tới các cơ sở dữ liệu Microsoft SQL Server.

1. Thiết lập kết nối với cơ sở dữ liệu:

***server = 'Admin'***

***database = 'QUANLI\_SACH'***

***username = 'TN'***

***password = '1407'***

***mydb=pyodbc.connect('DRIVER={SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+ password)***

* Đoạn mã này thiết lập các thông số kết nối tới cơ sở dữ liệu Microsoft SQL Server và sử dụng chúng để kết nối tới cơ sở dữ liệu đó.

1. Thực thi truy vấn và lấy dữ liệu:

***cursor = mydb.cursor()***

***query = "SELECT MASACH, TEN, SOLUONG, NAM\_XUATBAN, THELOAI, GIA FROM SACH"***

***cursor.execute(query)***

***rows = cursor.fetchall()***

* Đoạn mã này tạo ra một con trỏ để thực thi truy vấn SQL và thực hiện truy vấn để lấy các dòng dữ liệu từ cơ sở dữ liệu. Các dòng dữ liệu được lưu trữ trong biến rows.

1. Trích xuất dữ liệu từ kết quả truy vấn:

***TEN = [row[1] for row in rows]***

***GIA = [row[5] for row in rows]***

* Đoạn mã này trích xuất thông tin tên và giá của các quyển sách từ kết quả truy vấn. Dữ liệu này được lưu trữ trong các danh sách TEN và GIA.

1. Vẽ biểu đồ cột giá sách theo tên sách:
2. **Biểu đồ cột (Bar)**

***x\_pos = np.arange(len(TEN))***

***plt.bar(x\_pos, GIA, align='center', alpha=0.5)***

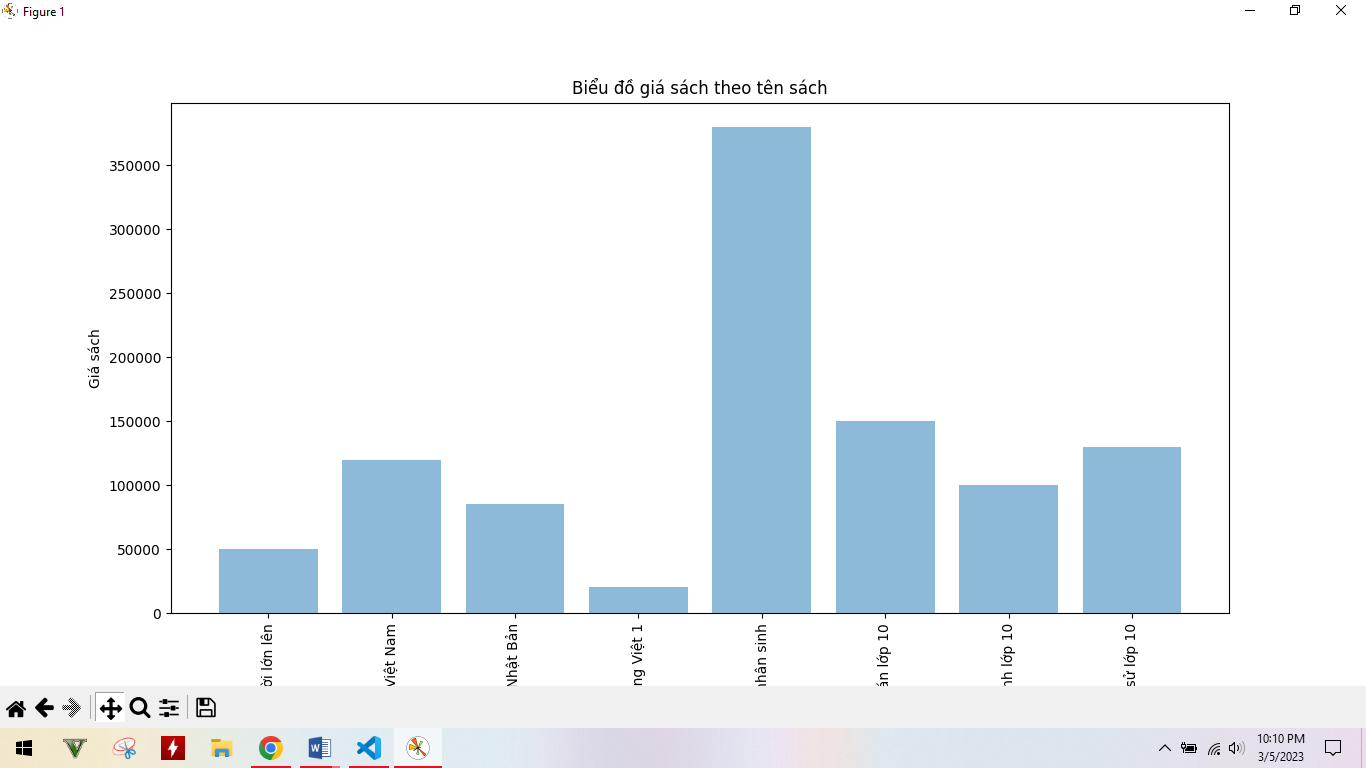
***plt.xticks(x\_pos, TEN, rotation=90)***

***plt.ylabel('Giá sách')***

***plt.title('Biểu đồ giá sách theo tên sách')***

***plt.show()***

* Đoạn mã này tạo ra một biến x\_pos chứa vị trí của các cột trên trục x. Biểu đồ cột được vẽ bằng lệnh plt.bar
* Dưới đây là biểu đồ:

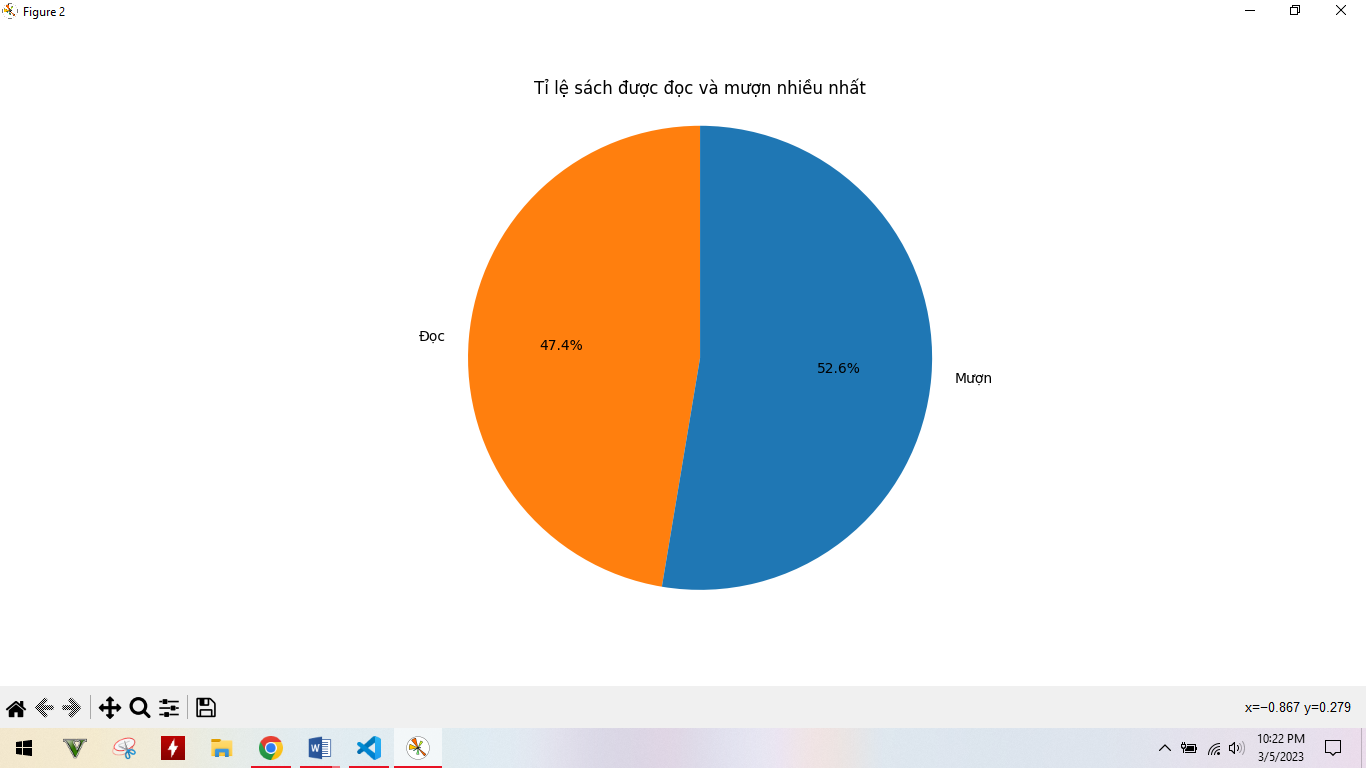
*Biểu đồ cột*

1. **Biểu đồ Tròn ( Pie )**

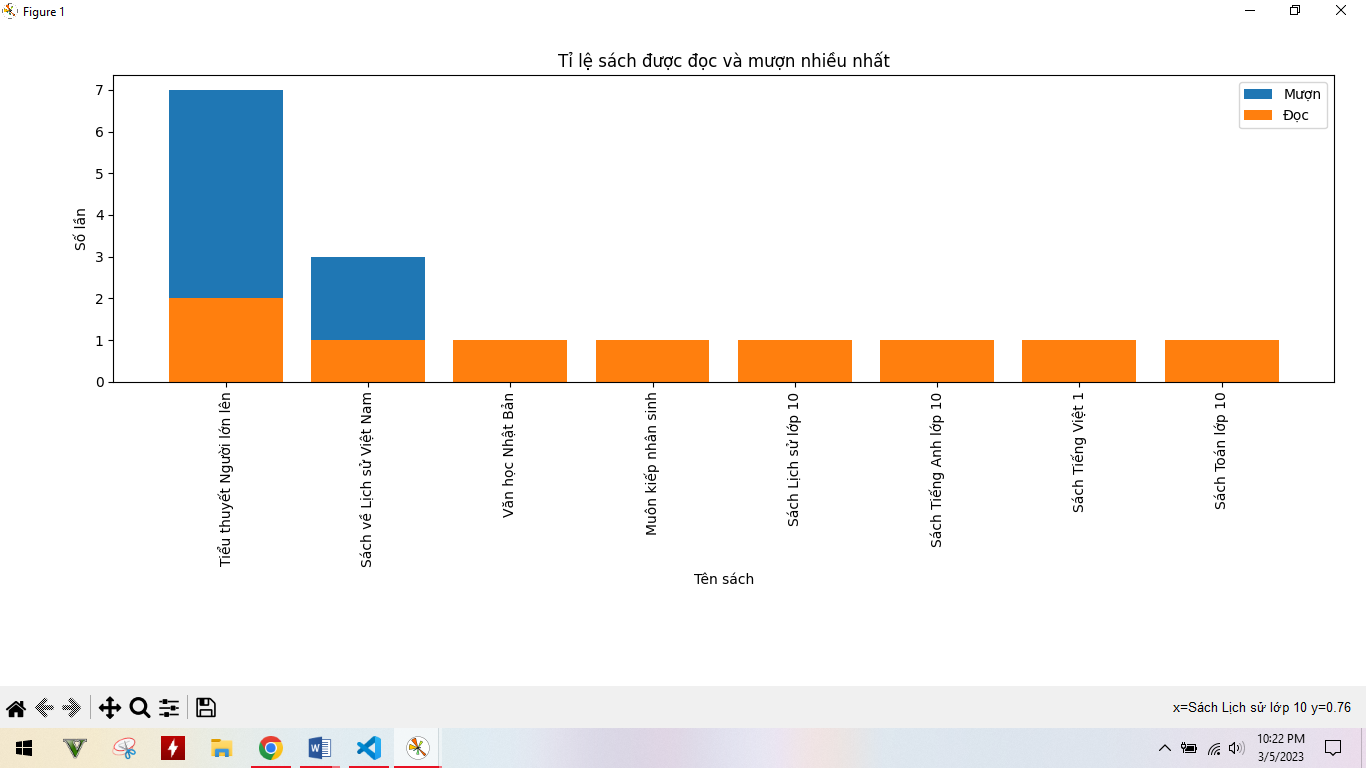
Đầu tiên, chúng ta sử dụng thư viện pyodbc để kết nối và truy vấn dữ liệu từ một cơ sở dữ liệu SQL Server. Sau đó, chúng ta sử dụng matplotlib để vẽ biểu đồ tròn:

1. Kết nối đến cơ sở dữ liệu SQL Server sử dụng pyodbc.connect.
2. Thực thi một truy vấn SQL để lấy dữ liệu từ bảng SACH, sử dụng cursor.execute và fetchall.
3. Sử dụng list comprehension để trích xuất dữ liệu từ kết quả truy vấn, lấy tên sách và giá của sách.
4. Sử dụng numpy.arange để tạo ra một dãy số nguyên liên tiếp từ 0 đến n-1, với n là số phần tử trong list TEN, sau đó sử dụng hàm plt.bar để vẽ biểu đồ cột. Dữ liệu trục hoành là tên sách và trục tung là giá của sách. Tiếp theo, chúng ta sử dụng plt.xticks để đặt tên các nhãn trục hoành của biểu đồ và plt.ylabel và plt.title để đặt tên trục tung và tiêu đề biểu đồ.
5. Sử dụng cursor.execute để thực thi truy vấn SQL khác để lấy số lần đọc và số lần mượn sách của từng cuốn sách.
6. Tính tổng số lần đọc và mượn sách, sau đó vẽ biểu đồ tròn sử dụng plt.pie, trong đó cung cấp tỉ lệ phần trăm số lần đọc và số lần mượn của từng cuốn sách và đặt tiêu đề biểu đồ bằng plt.title.
7. Sử dụng plt.show() để hiển thị biểu đồ.

Tiếp tục với đoạn code :

* Dòng đầu tiên sử dụng cursor để thực thi truy vấn SQL, lấy ra số lần đọc và số lần mượn cho mỗi cuốn sách từ bảng SACH và PHIEUMUONSACH.
* Dòng tiếp theo lấy kết quả truy vấn và lưu trữ nó vào biến result.
* Biến books chứa tên sách, biến reads chứa số lần đọc và biến borrows chứa số lần mượn tương ứng với từng cuốn sách.
* Tiếp theo, đoạn code vẽ biểu đồ cột sử dụng hàm bar trong thư viện Matplotlib. Các thanh cột đầu tiên biểu thị số lần mượn, thanh cột thứ hai biểu thị số lần đọc của từng cuốn sách.
* Đoạn code tiếp theo tính tổng số lần mượn và đọc sách.
* Biểu đồ tròn được vẽ bằng hàm pie. Biểu đồ này thể hiện tỉ lệ giữa số lần mượn sách và số lần đọc sách, với màu sắc được chỉ định bằng biến colors. Dưới đây là hình ảnh biểu đồ mô tả số lần sách đọc và mượn nhiều nhất:

*Biểu đồ tròn*



*Bảng chi tiết*

1. **Kết Luận**

Bài tiểu luận sau khi thực hiện phân tích và thiết kế hệ thống quản lý sách bằng Python và SQL Server, chúng tôi nhận thấy rằng đây là một giải pháp hiệu quả và tiện ích để quản lý các thông tin liên quan đến sách trong một thư viện hay cửa hàng sách. Hệ thống của chúng tôi sử dụng Python để tạo giao diện cho người dùng và quản trị viên quản lý thông tin của các cuốn sách. Python được chọn làm ngôn ngữ lập trình chính vì tính linh hoạt, dễ học và độ phổ biến cao. Đồng thời, hệ thống sử dụng SQL Server để lưu trữ, quản lý và truy xuất dữ liệu. SQL Server được chọn vì tính bảo mật cao, khả năng mở rộng và hiệu suất cao.

Hệ thống quản lý sách của chúng tôi cung cấp cho người dùng các chức năng như tìm kiếm, thêm mới, chỉnh sửa và xóa các thông tin liên quan đến sách. Người dùng cũng có thể tạo ra danh sách các cuốn sách yêu thích, giỏ hàng hoặc đánh giá các cuốn sách đã đọc. Ngoài ra, hệ thống của chúng tôi cũng cho phép quản trị viên quản lý các thông tin liên quan đến sách như tác giả, nhà xuất bản, danh mục sách, v.v. Quản trị viên cũng có thể thêm mới, sửa đổi hoặc xóa bất kỳ thông tin nào trong hệ thống.

Trong tương lai, hệ thống quản lý sách sẽ tiếp tục phát triển và cải tiến để đáp ứng nhu cầu ngày càng cao của người dùng. Hệ thống này sẽ được phát triển trên nền tảng công nghệ hiện đại và có tính linh hoạt cao để dễ dàng thích nghi với những thay đổi của thị trường.

Hệ thống quản lý nhà sách sử dụng Python và MySQL là một ví dụ cho việc sử dụng công nghệ hiện đại trong quản lý sách. Trong tương lai, các công nghệ này sẽ được cải tiến và phát triển để trở nên tối ưu hơn và đáp ứng được nhiều yêu cầu hơn. Hệ thống quản lý sách trong tương lai sẽ tích hợp nhiều tính năng mới nhằm đáp ứng nhu cầu của người dùng. Chẳng hạn như tính năng đặt mượn sách trực tuyến, cho phép người dùng tìm kiếm sách dễ dàng hơn, tính năng gợi ý sách dựa trên sở thích của người dùng, tính năng chia sẻ sách qua mạng xã hội và nhiều tính năng khác. Hệ thống quản lý sách trong tương lai cũng sẽ tích hợp trí tuệ nhân tạo để phân tích dữ liệu và đưa ra những gợi ý sách phù hợp với sở thích của từng người dùng. Hệ thống sẽ tự động tạo ra các bảng thống kê về việc mượn sách, sự quan tâm của người dùng đối với các sách và các thông tin khác giúp nhà sách cải thiện dịch vụ của mình.

Tóm lại, hệ thống quản lý sách bằng Python và SQL Server của chúng tôi là một giải pháp đáng tin cậy và tiện ích để quản lý thông tin liên quan đến sách. Hệ thống cung cấp các tính năng hiệu quả và đáp ứng được nhu cầu của người dùng và quản trị viên. Chúng tôi hi vọng rằng hệ thống của chúng tôi sẽ giúp các cửa hàng sách và thư viện quản lý sách của mình một cách hiệu quả hơn và tốt hơn.