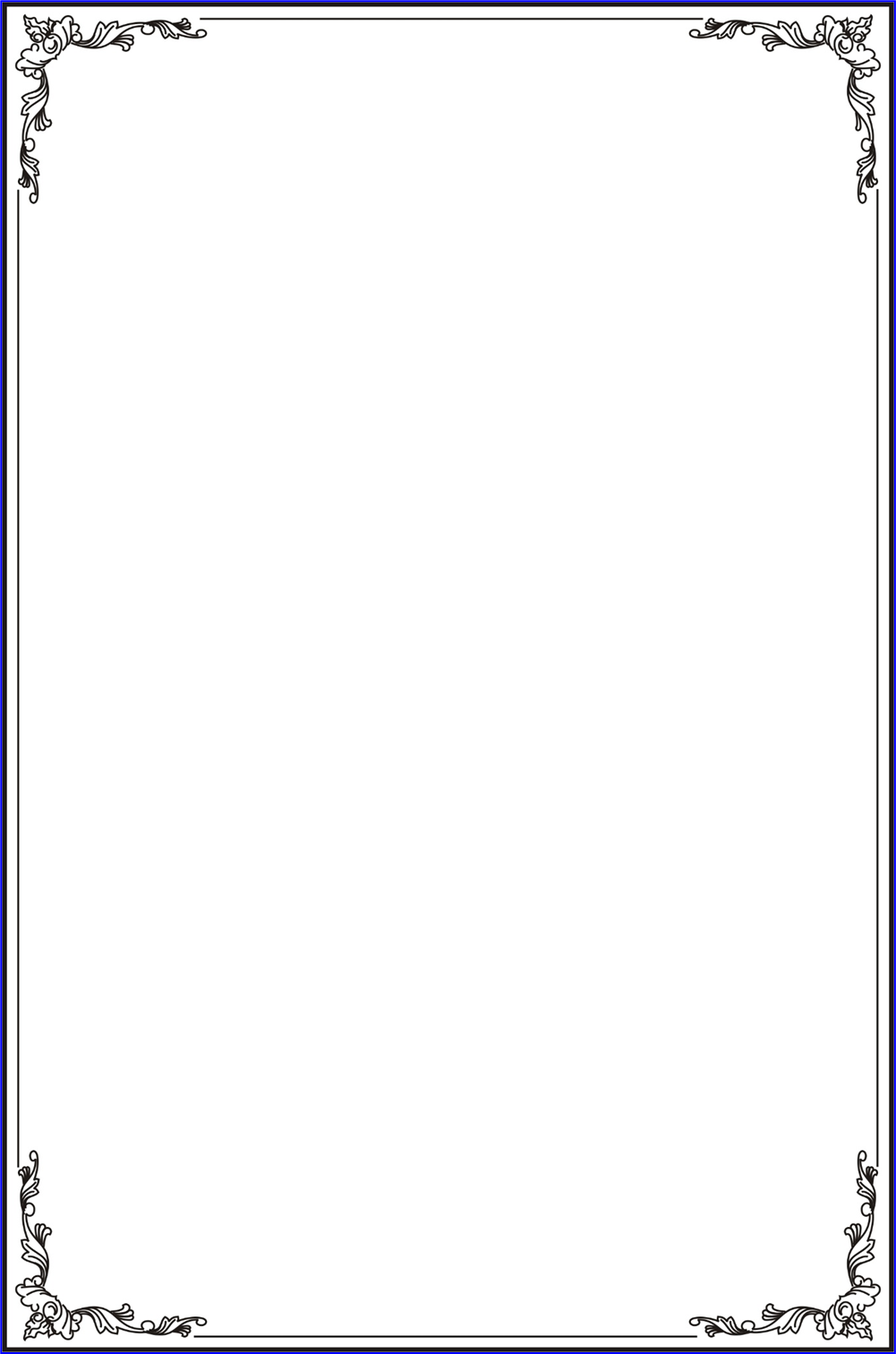
****

A picture containing text, clipart

Description automatically generated

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**

**Sinh viên thực hiện: Nhóm 9**

* **Đỗ Minh Dũng + 20110620 (NT)**
* **Bùi Tấn Đạt + 20133033**
* **Nguyễn Đức Thành + 20110307**
* **Nguyễn Thái Ngọc Tân + 20110719**

**Mã lớp học phần: DBMS330284\_22\_1\_02**

**Giảng viên hướng dẫn: Ts. Nguyễn Thành Sơn**

**TP. Hồ Chí Minh, tháng 11 năm 2022**

**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**DANH SÁCH THÀNH VIÊN THAM GIA**

**THỰC HIỆN ĐỀ TÀI VÀ VIẾT BÁO CÁO**

**Môn: Hệ quản trị cơ sở dữ liệu**

*HỌC KÌ I – NĂM HỌC 2022 – 2023*

***Nhóm số: 09***

***Tên đề tài: Xây dựng phần mềm quản lý cửa hàng bán sách***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** | **SỐ ĐIỆN THOẠI** |
| 1 | Đỗ Minh Dũng (NT) | 20110620 | 0376177512 |
| 2 | Bùi Tấn Đạt | 20133033 | 0397437703 |
| 3 | Nguyễn Đức Thành | 20110307 | 0342293128 |
| 4 | Nguyễn Thái Ngọc Tân | 20110719 | 0923140493 |

**Điểm: ……………………………………………………………………**

**Nhận xét của giảng viên:**

**……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

*Ngày tháng 11 năm 2021*

*Giảng viên chấm điểm*

*Ts. Nguyễn Thành Sơn*

# 

# LỜI MỞ ĐẦU

Ngày nay, Công nghệ thông tin đã và đang đóng vai trò quan trọng trong đời sống kinh tế, xã hội của nhiều quốc gia trên thế giới, là một phần không thể thiếu trong xã hội năng động và ngày càng hiện đại hoá. Công nghệ thông tin được ứng dụng vào hầu hết các lĩnh vực của đời sống và lĩnh vực hoạt động kinh doanh buôn bán không phải là một ngoại lệ. Việc ứng dụng công nghệ thông tin vào quản lý bán hàng không những tiết kiệm thời gian, tiết kiệm chi phí và nguồn nhân lực mà còn thể hiện được độ chính xác cao từ đó nâng cao hiệu quả công việc và tăng năng lực quản lý.

Xuất phát từ nhu cầu thực tế đó, trong công việc mua và bán sách, việc quản lý sách nhập và bán là một việc không thể thiếu. Nhằm thay thế việc quản lý bán hàng theo hình thức thô sơ, nhiều thủ tục, nhiều công đoạn,.. tại mỗi cửa hàng đạt hiệu quả không cao. Vì vậy, chúng em đã thực hiện báo cáo với đề tài “**Xây dựng hệ thống quản lý cửa hàng bán sách”.** Đây là một chương trình ứng dụng, do một bộ phận nhân viên trong nhà sách quản lý, với mục đích tiết kiệm thời gian và thuận tiện cho việc quản lý thông tin tại các nhà sách.

# 

# LỜI CẢM ƠN

Để hoàn thành đồ án cuối kỳ môn học này, cả nhóm xin tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Thành Sơn, đã tận tình hướng dẫn cả nhóm, chỉ dạy hết các kiến thức phù hợp trong suốt quá trình học để có thể thực hiện đề tài. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu đồ án mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Tuy nhiên trong quá trình nghiên cứu đề tài, mặc dù đã cố gắng hoàn thành tốt công việc được phân công trong phạm vi và khả năng cho phép nhưng chắc chắn không tránh khỏi những thiếu sót. Rất mong nhận được sự góp ý của thầy để chương trình được hoàn chỉnh hơn, đồng thời bổ sung vốn kinh nghiệm cho nhóm trên con đường sắp tới.

Nhóm em xin chân thành cảm ơn!

*Tp. HCM, 26,tháng 11 năm 2022*

*Nhóm Trưởng*

*Minh Dung*

*Đỗ Minh Dũng*

**DANH SÁCH PHÂN CÔNG VÀ SINH VIÊN THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** | **CÔNG VIỆC THỰC HIỆN** |
| 1 | Đỗ Minh Dũng (NT) | 20110620 | Thiết kế database, các trigger, procedure,funtion tương quan với table, chỉnh sửa lỗi databaes, diễn giải dữ liệu, thiết kế, cài đặt và thực hiện các giao diện chức năng:thêm sửa xóa voucher, account, phân quyền người dùng. |
| 2 | Bùi Tấn Đạt | 20133033 | Thiết kế database, các trigger, procedure,funtion tương quan với table, cài đặt và thực hiện các giao diện chức năng: thêm sửa xóa customer, typecustomer, author. Tìm kiếm và phân lọc dữ liệu cho chương trình, |
| 3 | Nguyễn Đức Thành | 20110307 | Tối ưu chương trình, lên ý tưởng cho chương trình. Thiết kế database, các trigger, procedure,funtion tương quan với table,cài đặt và thực hiện các giao diện chức năng: thêm sửa xóa các bảng hóa đơn nhập, hóa đơn xuất, thêm giỏ hàng, sản phẩm, tính năng nhập hàng, chi tiết đơn hàng. |
| 4 | Nguyễn Thái Ngọc Tân | 20110719 | Thiết kế database, các trigger, procedure,funtion tương quan với table, diễn giải dữ liệu, tìm kiếm và phân lọc dữ liệu cho chương trình, thiết kế, cài đặt và thực hiện các giao diện chức năng:thêm sửa xóa category, book-cate, book-author, thống kê doanh thu. |

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **CÁC TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| CSDL | Cơ sở dữ liệu |
| NXB | Nhà xuất bản |
|  |  |

**Mục lục**

[LỜI MỞ ĐẦU 2](#_Toc32084)

[LỜI CẢM ƠN 3](#_Toc13524)

**[A. TỔNG QUAN VỀ ĐỀ TÀI 5](#_Toc6104)**

[1. Giới thiệu đề tài 5](#_Toc17778)

[2. Mục tiêu của đề tài 6](#_Toc7968)

[3. Các chức năng chính của đề tài 6](#_Toc30547)

[4. Mô tả giao diện đề tài 7](#_Toc22770)

[5. Công nghệ được sử dụng 7](#_Toc26529)

**[B. NỘI DUNG 8](#_Toc2657)**

**[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 8](#_Toc14698)**

[1. Sơ lược về cơ sở dữ liệu và SQL Server 8](#_Toc7014)

[1.1 Khái quát cở sở dữ liệu 8](#_Toc18782)

[1.2 Hệ quản trị cơ sở dữ liệu 9](#_Toc8224)

[1.3 Sơ lược về SQL Server 10](#_Toc9354)

[2. Khái quát về Winforms, ADO.NET trong C# 12](#_Toc15661)

[2.1. Lập trình Winforms trong C# 12](#_Toc462)

[2.2 ADO.NET trong C# 13](#_Toc18366)

[3. Sơ lược về mô hình ba lớp 15](#_Toc15393)

[3.1 Khái niệm 15](#_Toc18257)

[3.2 Các thành phần chính 15](#_Toc31715)

[3.3 Ưu điểm và nhược điểm 15](#_Toc28216)

**[CHƯƠNG 2: PHÂN TÍCH , XÂY DỰNG DỮ LIỆU CỦA ĐỀ TÀI 17](#_Toc28618)**

[\*\* Sơ đồ ERD Tổng quát của đề tài: 17](#_Toc1875)

[1. Biểu diễn từ sơ đồ erd sang lược đồ quan hệ và mô tả thuộc tính 17](#_Toc14894)

[1.1 Bảng tác giả (AUTHOR) 17](#_Toc2730)

[1.2 Bảng nhà xuất bản (PUBLISHER) 18](#_Toc7616)

[1.3 Bảng Sách (BOOK) 18](#_Toc22475)

[1.4 Bảng tác giả của sách (BOOK\_AUTHOR)(Bảng rã) 19](#_Toc3316)

[1.5 Bảng thể loại (CATEGORY) 19](#_Toc19474)

[1.6 Bảng Thể loại của sách (BOOK\_CATEGORY)(bảng rã) 20](#_Toc6553)

[1.7 Bảng loại khác hàng (TYPECUSTOMER) 20](#_Toc23630)

[1.8 Bảng khách hàng (CUSTOMER) 21](#_Toc28291)

[1.9 Bảng nhân viên (EMPLOYEE) 21](#_Toc21591)

[1.10 Bảng phiếu quà tặng (VOUCHER) 22](#_Toc2865)

[1.11 Bảng hóa đơn nhập sách (BILLINPUT) 23](#_Toc13460)

[1.12 Bảng hóa đơn nhập theo sách (BOOK\_BILLINPUT)(bảng rã) 23](#_Toc31174)

[1.13 Bảng hóa đơn bán sách (BILLOUTPUT) 24](#_Toc14820)

[1.14 Bảng hóa đơn bán theo sách (BOOK\_BILLOUTPUT)(bảng rã) 25](#_Toc27354)

[1.15 Bảng tài khoản (ACCOUNT) 25](#_Toc12056)

[\*\* Mô hình Diagram của đề tài: 26](#_Toc7787)

[2. Mô tả khái quát các chức năng với các đối tượng 26](#_Toc6144)

[2.1 Đối với người quản lý 26](#_Toc8582)

[2.2 Đối với nhân viên 26](#_Toc31574)

[2.3 Đối với khách hàng 27](#_Toc16175)

[3. Mô tả các chức năng chi tiết của hệ thống 27](#_Toc8825)

[3.1. Chức năng quản lý nhân viên 27](#_Toc17418)

[3.2 Chức năng Quản lý khách hàng 27](#_Toc13624)

[3.3. Chức năng Quản lý kho sách 27](#_Toc30592)

[3.4 Chức năng Quản lý Voucher giảm giá 27](#_Toc26847)

[3.5 Chức năng Quản lý hóa đơn 28](#_Toc16138)

[3.6 Chức năng Quản lý phiếu nhập 28](#_Toc1249)

[3.7. Chức năng Quản lý doanh thu 28](#_Toc17606)

[3.80 Chức năng lập hóa đơn bán hàng 28](#_Toc20878)

[4. Sơ lược về các ràng buộc 28](#_Toc961)

[5. Cài đặt cơ sở dữ liệu và các ràng buộc tương ứng 29](#_Toc27060)

**[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG,CÁC CHỨC NĂNG 34](#_Toc28581)**

[\*\*\*Code kết nối cơ sở dữ liệu: 34](#_Toc28197)

[1. Danh sách các trigger của đề tài 37](#_Toc23201)

[1.1 Kiểm tra số lượng sách thêm trong giỏ hàng phải luôn có đủ trong kho 37](#_Toc21098)

[1.2 Kiểm tra số lượng sách trong kho có đủ để thêm vào BOOK\_BILLOUTPUT 38](#_Toc5062)

[1.3 Kiểm tra điều kiện voucher áp dụng vào đơn hàng 38](#_Toc29753)

[1.4 Kiểm tra điều kiện thỏa ràng buộc số điện thoại trong bảng AUTHOR 40](#_Toc18745)

[1.5 Kiểm tra điều kiện thỏa ràng buộc số điện thoại trong bảng PUBLISHER 40](#_Toc31585)

[1.6 Kiểm tra số điện thoại của khách hàng 40](#_Toc21723)

[1.7 Kiểm tra giá nhập sách phải bé hơn giá bán ra trong bảng BOOK 41](#_Toc11850)

[1.8 Kiểm tra phần trăm được giảm của khách hàng không lớn hơn 100% và không bé hơn 0% 41](#_Toc3932)

[1.9 Kiểm tra ngày bắt đầu phải nhỏ hơn hoặc bằng ngày kết thúc của VOUCHER 41](#_Toc1847)

[2. Danh sách các Function của đề tài 42](#_Toc25821)

[2.1 Hàm func\_checkVoucherValidOrNot 42](#_Toc30965)

[2.2 Hàm func\_checkPhone 42](#_Toc32640)

[2.3. Hàm func\_getAllCategoryOfBook 43](#_Toc1432)

[2.4 Hàm func\_getAllAuthorOfBook 43](#_Toc16280)

[2.5 Hàm func\_getIDCategory 43](#_Toc8300)

[2.6. Hàm func\_getIDAuthor 44](#_Toc13924)

[2.7. Hàm func\_returnIdBillFalse 44](#_Toc23636)

[2.8. Hàm func\_checkBookIsInBill 44](#_Toc16234)

[2.9. Hàm func\_getDataOfBillExport 45](#_Toc12836)

[2.10. Hàm func\_getDataOfBillImport 45](#_Toc30198)

[2.11. Hàm func\_getNameCusOfBillOutPut 45](#_Toc18101)

[2.12. Hàm func\_getNameEmpOfBillOutPut 45](#_Toc28488)

[2.13. Hàm func\_getNameBookById 46](#_Toc5837)

[2.14. Hàm func\_getAmountBookById 46](#_Toc10099)

[2.15. Hàm func\_returnToTalOfBillOutput 46](#_Toc24335)

[2.16. Hàm func\_returnToTalOfBillInput 47](#_Toc15873)

[2.17. Hàm func\_getIdEmployee 47](#_Toc2183)

[2.18. Hàm func\_getValueDiscountOfTypeCustomer 48](#_Toc18640)

[2.19. Hàm func\_searchAccount 48](#_Toc26264)

[2.20 Hàm func\_searchAuthor 48](#_Toc30407)

[2.21. Hàm func\_searchCategory 49](#_Toc14675)

[2.22. Hàm func\_searchEmployee 49](#_Toc18777)

[2.23. Hàm func\_searchCustomer 49](#_Toc16250)

[2.24. Hàm func\_searchPublisher 49](#_Toc8939)

[2.25. Hàm func\_searchReceiptExport 50](#_Toc15815)

[2.26. Hàm func\_searchVoucher 50](#_Toc6552)

[2.27. Hàm func\_searchTypeCustomer 50](#_Toc19751)

[2.28. Hàm func\_searchReceiptImport 50](#_Toc14568)

[2.29. Hàm func\_searchBook 51](#_Toc4103)

[3. Danh sách các Procedure của đề tài 51](#_Toc29304)

[3.1 Procedure thêm dữ liệu vào các bản 51](#_Toc12013)

[3.2 Procedure chỉnh sửa dữ liệu ở các bảng 58](#_Toc15523)

[3.3 Procedure xóa dữ liệu ở các bảng 61](#_Toc16112)

[3.4 Các chức năng thao tác nhập với sách 63](#_Toc6058)

[3.4.3. Procedure sửa số lượng sách trong đơn hàng 64](#_Toc2250)

[3.5 Các chức năng thao tác bán sách 66](#_Toc13553)

[3.6 Các thống kê doanh thu của cửa hàng 71](#_Toc7574)

[3.6.2. Procedure show top 5 sản phẩm bán chạy nhất trong 2 mốc thời gian do người dùng nhập 71](#_Toc6472)

[4. Danh sách các View của đề tài 72](#_Toc20097)

[4.1. View hiển thị ra thông tin tài khoản 72](#_Toc6746)

[4.2. View hiển thị thông tin tác giả 72](#_Toc1777)

[4.3. View hiển thị thông tin nhà xuất bản 72](#_Toc22712)

[4.5. View hiển thị thông tin sách 72](#_Toc17645)

[4.6. View hiển thị thông tin thể loại 72](#_Toc8989)

[4.7. View hiển thị thông tin hóa đơn nhập 72](#_Toc2515)

[4.8. View hiển thị thông tin hóa đơn xuất 73](#_Toc21716)

[4.9. View hiển thị thông tin khách hàng 73](#_Toc11670)

[4.10. View hiển thị tất cả thông tin loại khách hàng 73](#_Toc4211)

[4.11. View hiển thị toàn bộ thông tin nhân viên 73](#_Toc4381)

[4.12. View hiển thị toàn bộ thông tin voucher 73](#_Toc327)

[4.13. View hiển thị danh sách đang bán trong cửa hàng 73](#_Toc20610)

[4.14 View hiển thị doanh thu top 5 sản phẩm bán chạy nhất 73](#_Toc24856)

[5. Tạo user và phân quyền viết dưới dạng procedure 74](#_Toc23603)

[5.1. Procedure tạo user 74](#_Toc11578)

[5.2. Procedure cập nhật tài khoản user 74](#_Toc22606)

[5.3. procedure xóa user 74](#_Toc3336)

[5.4. procedure phân quyền cho admin và nhân viên 75](#_Toc23469)

**[CHƯƠNG 4: THIẾT KẾ GIAO DIỆN WINFORM 77](#_Toc15665)**

[1. Giao diện form đăng nhập 77](#_Toc13437)

[1.1. Đăng nhập với quyền admin 77](#_Toc11910)

[1.2. Đăng nhập với quyền nhân viên 78](#_Toc18000)

[1.3. Các quyền cấm nhân viên không được truy cập vào 79](#_Toc15407)

[2. Giao diện bảng AUTHOR 80](#_Toc30223)

[2.1 Thêm tác giả 80](#_Toc15459)

[3. Giao diện form Bảng PUBLISHER 81](#_Toc31292)

[4. Giao diện bảng CATEGORY 81](#_Toc2081)

[5. Giao diện Bảng BOOK 82](#_Toc23125)

[6. Giao diện bảng EMPLOYEE 82](#_Toc14717)

[7. Giao diện bảng VOUCHER 83](#_Toc32358)

[8. Giao diện bảng TYPECUSTOMER 83](#_Toc7545)

[9. Giao diện bảng CUSTOMER 84](#_Toc30725)

[10. Giao diện bảng BILLINPUT 84](#_Toc32134)

[11. Giao diện bảng ACCOUNT 85](#_Toc14409)

[12. Giao diện chức năng thống kê doanh thu 86](#_Toc17946)

**[C. KẾT LUẬN 87](#_Toc17034)**

[1. Kết quả đạt được 87](#_Toc31495)

[1.1 Về lý thuyết 87](#_Toc10713)

[1.2 Về đề tài 87](#_Toc12148)

[2. Hạn chế 87](#_Toc22429)

[3. Hướng phát triển 88](#_Toc9892)

**[D. TÀI LIỆU THAM KHẢO 89](#_Toc19385)**

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu đề tài

Từ xưa đến nay, sách luôn được coi là một người bạn gắn bó của con người, có lẽ bởi vì sách có vai trò hết sức to lớn với đời sống nhân loại. Sách đơn giản là một nơi lưu trữ tri thức của nhân loại, đó là nơi ghi lại những dấu ấn lịch sử, là nơi để con người có thể học tập và nuôi dưỡng tâm hồn. Thuở sơ khai, sách được làm bằng tre, trúc, bằng lụa, vải, gần hơn chúng ta biết sách được làm bằng giấy, hiện đại hơn thì có e+ book – sách điện tử. Và dù tồn tại dưới bất kì hình thức nào thì sách vẫn mang trong mình những vai trò quan trọng không thể chối bỏ.

Một chủ cửa hàng bán sách cần làm các công việc như quản lý nhân viên, quản lý sách, khách hàng thành viên, mối quan hệ với nhà cung cấp, đều chỉ mang cái nhìn tổng quát về công việc kinh doanh của cửa hàng. Người quản lý cửa hàng rất khó khăn trong việc nắm bắt các thông tin về nhà cung cấp, khách hàng, số lượng sách đã bán được, số lượng sách còn tồn trong kho, loại sách nào bán chạy để có thể đưa ra biện pháp kinh doanh kịp thời. Mỗi khi muốn thống kê, báo cáo về hoạt động kinh doanh của cửa hàng, các nhân viên lại phải tìm tòi, lùng sục từng hóa đơn, sổ sách mất nhiều thời gian, dẫn đến sự chậm trễ trong việc đưa ra chiến lược kinh doanh.

Trong hoạt động kinh doanh, sản xuất, mua bán ngày nay việc quản lý của mỗi cửa hàng về các hoạt động mua bán sản xuất và ứng dụng công nghệ thông tin đã khẳng định được vai trò xúc tác và thúc đẩy sự phát triển của các cửa hàng, doanh nghiệp lớn, vừa và nhỏ. Đối với cửa hàng, một hiệu sách, việc quản lý các thông tin của cửa hàng mình như kho hàng, thông tin nhân viên,.. cũng cần phải dễ dàng và dễ quản lý thay vì mô hình quản lý thủ công như trước . Người chủ cửa hàng hoặc nhân viên sẽ dễ quản lý các thông tin của cửa hàng sách đó. Nó còn giúp người chủ cửa hàng dễ dàng nắm bắt, quản lí sản phẩm của mình và quảng cáo sản phẩm của mình được nhiều người biết đến hơn, đồng thời cạnh tranh cao với các cửa hàng khác, sẽ giúp cửa hàng trở nên dễ dàng và tiện lợi hơn rất nhiều so với việc làm thủ công bằng tay và giấy tờ.

Do đó để khai thác ý tưởng này đặc biệt này, việc xây dựng **“Ứng dụng quản lý cửa hàng bán sách”** (cho người quản lý)là một đề tài vô cùng thực tế, có thể ứng dụng được và có tiềm năng phát triển trong tương lai. Và nhóm em chọn làm đề tài cho dự án cuối kỳ này.

## 2. Mục tiêu của đề tài

Xây dựng một hệ thống cho các cửa hàng nói chung có thể quản lý hàng hóa một cách đơn giản và hiệu quả hơn. Xây dựng một hệ thống quản lý đơn giản, nhanh gọn, chính xác và tốn ít thời gian, công sức nhất;

Các mục tiêu quan trọng của đề tài:

* Tính chính xác: Mọi thông tin của mọi cuốn sách phải đảm bảo chính xác 100% để không ảnh hưởng đến kết số lượng sách, giá sách....
* Tính nhanh gọn: việc tìm thông tin cũng như xử lý thông tin phải nhanh gọn, tiết kiệm thời gian
* Tính hiệu quả: tìm kiếm phải đạt tốc độ cao. Chỉnh sửa đơn giản, hiệu quả. Giao diện đẹp, dễ thực hiện.
* Tính bảo mật: Độ bảo mật về thông tin cá nhân của phải cao. Hệ thống phải có hành vi ngăn chặn khi có sự xâm nhập bất hợp pháp từ bên ngoài.
* Tính ổn định: Hệ thống phải làm việc trơn chu, tránh sai sót về mặt dữ liệu và phải chiếm ít tài nguyên của máy.

## 3. Các chức năng chính của đề tài

Chương trình gồm có các chức năng chính:

+ Quản lý đăng nhập;

+ Quản lý nhân viên;

+ Quản lý khách hàng;

+ Quản lý kho sách;

+ Quản lý mã giảm giá;

+ Quản lý hóa đơn;

+ Quản lý phiếu nhập;

+ Lập hóa đơn bán hàng;

+ Thống kê doanh thu;

## 4. Mô tả giao diện đề tài

+ Giao diện đăng nhập: Gồm có Username và Password. Người dùng (người quản lý, nhân viên) khi muốn đăng nhập vào hệ thống cần có một tài khoản.

+ Giao diện cho nhân viên: Gồm các from Lập hóa đơn bán hàng, Quản lý khách hàng thành viên, Quản lý kho sách, Quản lý mã giảm giá, Quản lý hóa đơn, Quản lý phiếu nhập.

+ Giao diện cho người quản lý: Ngoài các from Lập hóa đơn bán hàng, Quản lý khách hàng, Quản lý kho sách, Quản lý mã giảm giá, Quản lý hóa đơn, Quản lý phiếu nhập giống như của nhân viên thì còn có from Quản lý nhân viên và Quản lý thu nhập.

## 5. Công nghệ được sử dụng

* Công nghệ ADO.NET
* Microsoft SQL Server Management Studio 18
* Windows Forms Application trên Microsoft Visual Studio 2019.

# NỘI DUNG

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Sơ lược về cơ sở dữ liệu và SQL Server

### 1.1 Khái quát cở sở dữ liệu

Một CSDL (Database) là một tập hợp các dữ liệu có liên quan với nhau, chứa thông tin của một tổ chức nào đó (như một trường học, một ngân hàng, một công ty, một nhà máy,…) được lưu trữ trên các thiết bị nhớ để đáp ứng nhu cầu khai thác thông tin của nhiều người sử dụng với nhiều mục đích khác nhau.

#### 1.1.1 Các mức thể hiện của cở sở dữ liệu

Mức vật lý: CSDL vật lý của một hệ CSDL là tập hợp các tệp dữ liệu, tồn tại trên các thiết bị nhớ.

Mức khái niệm: Những người quản trị hệ CSDL không cần hiểu chi tiết ở mức vật lí nhưng họ cần phải biết những dữ liệu nào được lưu trong CSDL. Giữa những dữ liệu có các mối liên hệ nào,… CSDL có thể mô tả như một bảng, mỗi cột là một thuộc tính, mỗi hàng tương ứng với dữ liệu về một chủ thể. Mức hiểu CSDL như vậy gọi là mức khái niệm

Mức khung nhìn: Khi khai thác CSDL, một người dùng có thể không quan tâm đến toàn bộ thông tin chứa trong CSDL mà chỉ cần một phần thông tin nào đó phù hợp với nghiệp vụ hay mục đích sử dụng của mình. Thể hiện phù hợp của CSDL cho mỗi người dùng được coi là một khung nhìn của CSDL. Mức hiểu CSDL của người dùng thông qua khung nhìn được gọi là mức khung nhìn.

#### 1.1.2 Các yêu cầu cơ bản của cở sở dữ liệu

CSDL phải đảm bảo các tính chất sau:

+ Tính cấu trúc: Dữ liệu trong CSDL được lưu trữ theo một cấu trúc xác định. Ví dụ, CSDL lớp có cấu trúc là bảng 50 dòng, 10 cột. Mỗi cột là một thuộc tính và mỗi dòng là một hồ sơ học sinh.

+ Tính toàn vẹn: Các giá trị dữ liệu được lưu trữ trong CSDL phải thỏa mãn một số ràng buộc, tùy thuộc vào hoạt động của tổ chức mà CSDL phản ánh.

+ Tính nhất quán: Sau những thao tác cập nhật dữ liệu và ngay cả khi có sự cố xảy ra (do phần cứng hay phần mềm) trong quá trình cập nhật, dữ liệu trong CSDL phải luôn đảm bảo tính đúng đắn.

+ Tính an toàn và bảo mật thông tin: CSDL vẫn được bảo vệ an toàn, phải ngăn chặn được những truy xuất không được phép và phải khôi phục được CSDL khi có sự cố ở phần cứng hay phần mềm.

+ Tính độc lập: Vì một CSDL phải phục vụ cho nhiều mục đích khai thác khác nhau nên dữ liệu phải độc lập với các ứng dụng, không phụ thuộc vào một vài bài toán cụ thể.

+ Tính không dư thừa: CSDL thường không lưu trữ những dữ liệu trùng lặp hoặc những thông tin có thể dễ dàng suy diễn hay tính toán được từ những dữ liệu đã có. Sự trùng lặp thông tin vừa làm lãng phí bộ nhớ để lưu trữ vừa dễ dẫn đến tình trạng không nhất quán thông tin.

### 1.2 Hệ quản trị cơ sở dữ liệu

#### 1.2.1 Định nghĩa Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu (Database Management System + DBMS): Là một hệ thống phần mềm cho phép tạo lập cơ sở dữ liệu và điều khiển mọi truy cập đối với cơ sở dữ liệu đó.

Trên thị trường phần mềm hiện nay ở Việt Nam đã xuất hiện khá nhiều phần mềm hệ quản trị cơ sở dữ liệu như: Microsoft Access, Foxpro, DB2, SQL Server, Oracle,…

Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System + RDBMS) là một hệ quản trị cơ sở dữ liệu theo mô hình quan hệ.

#### 1.2.2 Chức năng của Hệ quản trị cơ sở dữ liệu

+ Quản lý lượng dữ liệu lớn hiệu quả, quản lý những cơ sở dữ liệu tồn tại lâu dài, hỗ trợ quan sát dữ liệu

+ Hỗ trợ định nghĩa cấu trúc dữ liệu và thao tác trên dữ liệu thông qua các ngôn ngữ cao cấp nhất định

+ Cung cấp các truy cập chính xác cùng lúc với nhiều truy cập dữ liệu khác nhau.

+ Đảm bảo tính độc lập và bất biến của dữ liệu dù cấu trúc mô hình dữ liệu có thay đổi.

+ Bảo vệ dữ liệu khỏi các truy cập không được cấp phép

+ Phục hồi dữ liệu, tránh làm mất dữ liệu trên hệ thống

#### 1.2.3 Vai trò của Hệ quản trị cơ sở dữ liệu

Cung cấp môi trường tạo lập cơ sở dữ liệu: Vai trò của các hệ quản trị cơ sở dữ liệu đó chính là cung cấp cho người dùng một ngôn ngữ định nghĩa dữ liệu để cho họ có thể tự mô tả, khai báo kiểu và cấu trúc của dữ liệu

Cung cấp cách cập nhật và khai thác dữ liệu: Các hệ quản trị cơ sở dữ liệu sql có nhiệm vụ cung cấp cho người dùng ngôn ngữ thao tác dữ liệu để họ có thể diễn tả các yêu cầu cũng như thực hiện các thao tác cập nhật, khai thác CSDL. Những thao tác dữ liệu mà người dùng có thể thực hiện là nhập, sửa và xóa dữ liệu. Còn các khai thác dữ liệu bao gồm tìm kiếm và kết xuất dữ liệu

Cung cấp các công cụ kiểm soát và điều khiển các truy cập vào cơ sở dữ liệu: Đây là một trong các vai trò rất quan trọng của hệ quản trị cơ sở dữ liệu. Mục đích là để đảm bảo thực hiện một số yêu cầu cơ bản của hệ quản trị cơ sở dữ liệu. Các yêu cầu đó gồm:

* Đảm bảo an ninh và kịp thời phát hiện, thực hiện các hành động để ngăn chặn sự truy cập bất hợp phát;
* Duy trì được tính nhất quán của dữ liệu;
* Tổ chức, điều khiển các truy cập dữ liệu;
* Có thể khôi phục lại được cơ sở dữ liệu khi xảy ra các sự cố về phần cứng hoặc phần mềm;
* Quản lý các mô tả dữ liệu.

### 1.3 Sơ lược về SQL Server

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

#### 1.3.1. Lịch sử ra đời và các ấn bản của SQL server

Năm 1989, phiên bản đầu tiên của SQL Server 1.0 ra đời được dùng cho các hệ điều hành 16 bit và được phát triển cho tới ngày nay.

Cho tới khi SQL Server ra phiên bản 6.5 thì được thị trường chấp nhận rộng rãi. Một đột phá cải tiến cho SQL Server 7.0 khi được Microsoft viết lại một engine hoàn toàn mới. Đến khi SQL Server từ phiên bản 7.0 cải tiến lên 8.0 chủ yếu phát triển về tính năng thiết kế website.

Một vài ấn bản SQL Server:

+ Enterprise: là một ấn bản chứa tất cả các đặc điểm nổi bật của SQL Server như: các công cụ cho phép tạo và quản lý phân cụm SQL Server, nhân bộ máy cơ sở dữ liệu và một số dịch vụ đi kèm. Nó có thể đánh địa chỉ 12 terabytes và quản lý cơ sở dữ liệu lên tới 524 petabytes.

+ Standard: Ấn bản này có thể chạy tốt trên hệ thống lên tới 4 CPU và 2 GB RAM rất thích hợp cho các dịch vụ thiết kế web vừa và nhỏ.

+ Developer: Ấn bản này giới hạn số lượng người kết nối với server nhưng có đầy đủ các tính năng của Enterprise Edition. Đây là phiên bản được sử dụng cho kiểm tra và phát triển ứng dụng phù hợp cho các cá nhân trong lĩnh vực web như: freelancer Việt Nam, website spa,…

+ Workgroup: ấn bản SQL Server này có các chức năng lõi cơ sở dữ liệu nhưng không đi kèm các dịch vụ. Ở phiên bản 2012 không có ấn bản này.

+ Express: Ấn bản này dễ dàng sử dụng và quản trị cơ sở dữ liệu đơn giản.

#### 1.3.2 Các thành phần cơ bản trong SQL Server

Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,… Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

*+ Database Engine:* Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh

Ví dụ: trả lại tài nguyên cho hệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

*+ Integration Services:* Đây là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Mongo DB, Oracle, SQL Server, DB2, Microsoft Access,… và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

*+ Analysis Services:* Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Công cụ này giúp việc phân tích dữ liệu hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.

*+ Notification Services:* Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.

*+ Reporting Services:* Là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

*+ Full Text Search Service:* Là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

*+ Service Broker:* Là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

## Khái quát về Winforms, ADO.NET trong C#

### 2.1. Lập trình Winforms trong C#

Winform là một ứng dụng có giao diện người dùng đồ họa (GUI) nằm trong .Net framework. Mục đích của nó là cung cấp một giao diện đồ họa để phát triển các ứng dụng cho máy tính dễ dàng hơn. Winforms (Windows Form) chỉ có thể sử dụng để phát triển các ứng dụng chạy trên Desktop chứ không thể chạy trên web. Các ứng dụng WinForms có thể chứa các loại điều khiển khác nhau như label, textbox, listbox,…

Winform cho phép lập trình các ứng dụng Windows nhờ vào tính tiện ích, dễ viết, có giao diện kéo thả đơn giản,… Các GUI được thiết kế bằng việc thêm vào Form các điều khiển phù hợp.

\*\* Vai trò của WinForm:

* Chứa các điều khiển
* Xử lý dữ liệu người dùng nhập
* Kết nối với cơ sở dữ liệu
* Hiển thị thông tin

\*\* Ưu điểm:

* Tốc độ xử lý nhanh
* Đảm bảo bảo mật, an toàn thông tin
* Có thể tương thích với nhiều phiên bản Windows khác nhau
* Thao tác được trên nhiều giao diện

\*\* Nhược điểm:

* Phần mềm chạy trên Windows nên muốn sử dụng phải cài đặt phần mềm trên máy tính
* Chỉ sử dụng được trên Desktop
* Đồ họa không cao nên giao diện phần mềm thiếu tính trực quan

### 2.2 ADO.NET trong C#

#### 2.2.1. Giới thiệu về công nghệ ADO.NET

ADO.NET là một phần của .NET Framework, nó được xem là “bộ thư viện lớp” chịu trách nhiệm xử lý dữ liệu trong ngôn ngữ MS.NET. ADO.NET được thiết kế với dạng dữ liệu “ngắt kết nối”, nghĩa là chúng ta có thể lấy cả một cấu trúc phức tạp của dữ liệu từ database, sau đó ngắt kết nối với database rồi mới thực hiện các thao tác cần thiết. Đây là một sự tiến bộ về mặt thiết kế bởi vì thiết kế ADO trước đây luôn cần duy trì một kết nối trong quá trình thao tác dữ liệu.

#### 2.2.2. Kiến trúc ADO.NET

Kiến trúc ADO.NET gồm có hai phần:

Phần kết nối: được sử dụng khi ta kết nối với Database và thao tác dữ liệu, yêu cầu phải thực hiện kết nối với Database khi đang thao tác. \*\*Các đối tượng của phần kết nối là:

*+ Connection:* Đối tượng quản lý đóng /mở kết nối tới Database. Có hai dạng Connection tuỳ theo nguồn dữ liệu là gì (OleDb hay SQL Server) đó là OleDbConnection và SqlConnection.

*+ Command:* Đối tượng thực hiện các câu lệnh tương tác truy vấn, rút trích dữ liệu từ database khi đã thiết lập kết nối tới dữ liệu và trả về kết quả. Tương tự như Connection, Command cũng có hai dạng tùy theo nguồn dữ liệu là gì (OleDb hay SQL Server) đó là OleDbCommand và SqlCommand.

*+ DataReader:* Đối tượng xử lý đọc dữ liệu, được thiết kế phù hợp cho các ứng dụng web. Chỉ xử lý một dòng dữ liệu tại một thời điểm. Phù hợp với ứng dụng web vì xử lý nhanh, nhẹ không chiếm bộ nhớ. Cũng có 2 dạng tùy theo dữ liệu nguồn: OleDbDataReader và SqlDataReader. Dữ liệu của đối tượng được tạo ra khi đối tượng Command thực hiện câu lệnh ExecuteReader().

*+ DataAdapter:* Đây là đối tượng rất quan trọng của ADO.NET, nó là cầu nối của database và dataset (dataset là đối tượng ngắt kết nối), bởi vì đối tượng “ngắt kết nối” dataset không thể liên lạc trực tiếp với database nên nó cần một đối tượng trung gian lấy dữ liệu từ database cho nó. Và đó chính là DataAdapter. Vì DataAdapter khi thao tác với Database vẫn phải duy trì kết nối nên nó được liệt kê vào dạng “kết nối”, nhưng bản chất là phục vụ cho việc “ngắt kết nối”.

Phần “ngắt kết nối”: chỉ có một đối tượng chịu trách nhiệm ngắt kết nối đó chính là DataSet. DataSet không cần biết gì về Database thuộc kiểu gì, kết nối ra sao. Nhiệm vụ của DataSet là nhận dữ liệu về từ DataAdapter và xử lý nó. DataSet có thể được xem như 1 Database trong bộ nhớ gồm tất cả các bảng, quan hệ... DataSet có nhiều đối tượng được xem là “con” tức là cấp thấp hơn đi kèm với nó như: DataTable (tương đương với 1 bảng trong database), cấp thấp hơn của DataTable có các đối tượng DataRow (tương đương với 1 dòng), DataColumn( tương đương với 1 cột), DataRelation (tương đương với các quan hệ). Ngoài ra còn có các đối tượng nhóm: vd DataTableCollection, DataRowCollection, DataColumnCollection.

Việc sử dụng DataSet là một tiến bộ lớn của kiến trúc ADO.NET tuy nhiên với các ứng dụng Web, việc sử dụng DataSet không được khuyến khích vì đối tượng DataSet được xem là quá lớn, nặng nề khó thích hợp cho đường truyền trên web vốn rất hạn chế.

## Sơ lược về mô hình ba lớp

### 3.1 Khái niệm

Mô hình 3 lớp hay còn được gọi là mô hình Three Layer(3+Layer), mô hình này được phát triển nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo.

### 3.2 Các thành phần chính

Mô hình 3+layer gồm có 3 phần chính:

+ *Presentation Layer (GUI):* Lớp này có nhiệm vụ chính là giao tiếp với người dùng. Nó gồm các thành phần giao diện (winform, webform,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

*+ Business Logic Layer (BLL)*: Layer này phân ra 2 thành nhiệm vụ:

Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.

+ Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

*Data Access Layer (DAL):* Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,…).

### 3.3 Ưu điểm và nhược điểm

**Ưu điểm: +** Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và duy trì project tốt hơn

+ Dễ dàng phân loại các hành động tại Business

+ Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,…

+ Ứng dụng được cho các project lớn ở bên ngoài

+ Thao tác thay đổi dữ liệu có thể được cô lập trong 1 lớp hoặc chỉ ảnh hưởng đến lớp gần nhất. Điều đó đảm bảo không ảnh hưởng đến cả chương trình, tiết kiệm thời gian công sức.

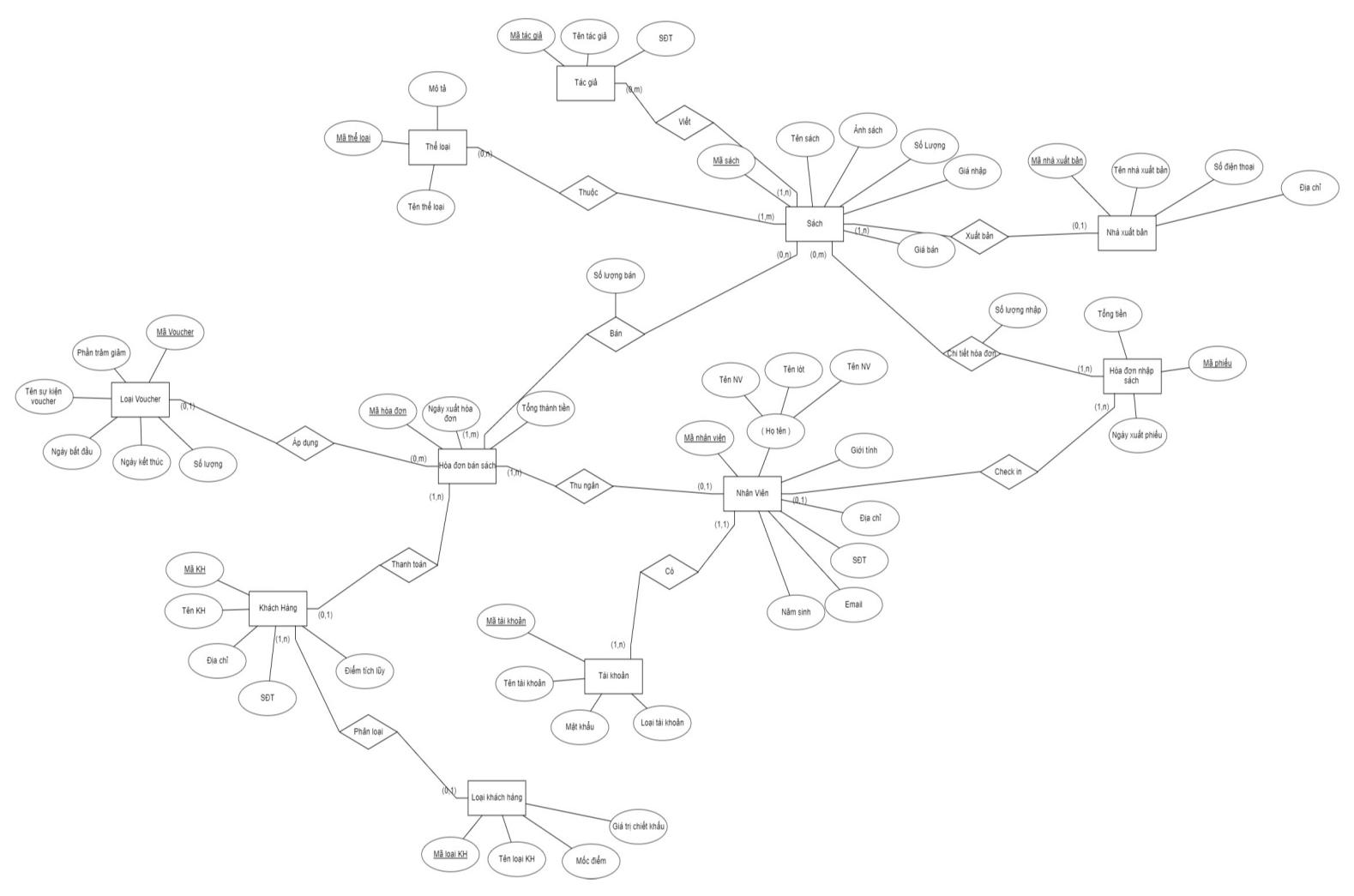
**Nhược điểm:**

+ Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói +> truyền đi +> mở gói trước khi có thể dùng được

+ Việc phát triển ứng dụng phức tạp hơn.

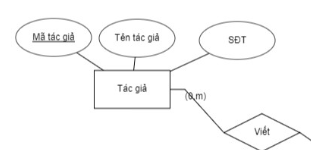
## CHƯƠNG 2: PHÂN TÍCH , XÂY DỰNG DỮ LIỆU CỦA ĐỀ TÀI

## \*\* Sơ đồ ERD Tổng quát của đề tài:



## Biểu diễn từ sơ đồ erd sang lược đồ quan hệ và mô tả thuộc tính

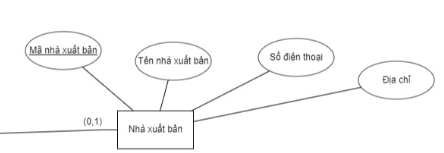
### 1.1 Bảng tác giả (AUTHOR)



AUTHOR(idAuthor,nameAuthor,phoneNumber)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idAuthor | varchar(8) | Mã tác giả | Khoá chính, not null |
| 2 | nameAuthor | nvarchar(30) | Tên tác giả | Not null, unique |
| 3 | phoneNumber | varchar (10) | Sđt tác giả |  |

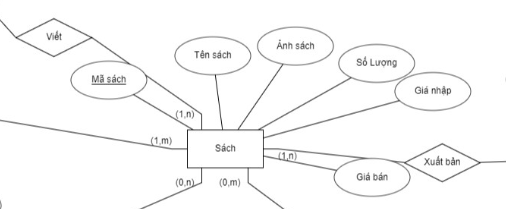
### 1.2 Bảng nhà xuất bản (PUBLISHER)



PUBLISHER(idPublisher,namePublisher,addressPublisher,phoneNumber)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idPublisher | varchar(8) | Mã nhà xuất bản | Khoá chính, not null |
| 2 | namePublisher | nvarchar(30) | Tên nhà xuất bản | Not null |
| 3 | addressPublisher | nvarchar(30) | Địa chỉ nhà xuất bản |  |
| 4 | phoneNumber | varchar (10) | Số điện thoại NXB |  |

### 1.3 Bảng Sách (BOOK)



BOOK(idBook, nameBook, urlImage, amount, priceImport, priceExport, IdPublisher)

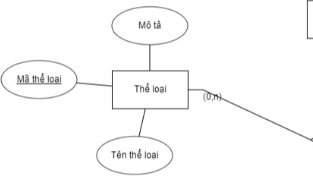
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBook | varchar(8) | Mã Sách | Khoá chính, not null |
| 2 | nameBook | nvarchar(20) | Tên Sách | Not null |
| 3 | urlImage | image | Ảnh Sách | Not null |
| 4 | amount | int | Số lượng sách | Not null |
| 5 | priceImport | int | Giá nhập |  |
| 6 | priceExport | int | Giá bán |  |
| 7 | idPublisher | varchar(8) | Mã nhà xuất bản | Khoá ngoại |

### 1.4 Bảng tác giả của sách (BOOK\_AUTHOR)(Bảng rã)

BOOK\_AUTHOR(idBook, idAuthor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBook | varchar(8) | Mã Sách | Khoá chính, khóa ngoại, not null |
| 2 | idAuthor | varchar(8) | Mã tác giả | Khóa chính, khóa ngoại, not null |

### 1.5 Bảng thể loại (CATEGORY)



CATEGORY(idCategory, nameCategory, describeCategory)

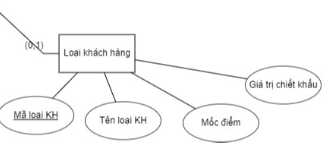
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idCategory | varchar(8) | Mã thể loại | Khoá chính, not null |
| 2 | nameCategory | nvarchar(20) | Tên thể loại | Unique, not null |
| 3 | describeCategory | nvarchar(50) | Mô tả thể loại |  |

### 1.6 Bảng Thể loại của sách (BOOK\_CATEGORY)(bảng rã)

BOOK\_CATEGORY(idBook, idCategory)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBook | varchar(8) | Mã Sách | Khoá chính, khóa ngoại, not null |
| 2 | idCategory | varchar(8) | Mã Thể Loại | Khoá chính, khóa ngoại, not null |

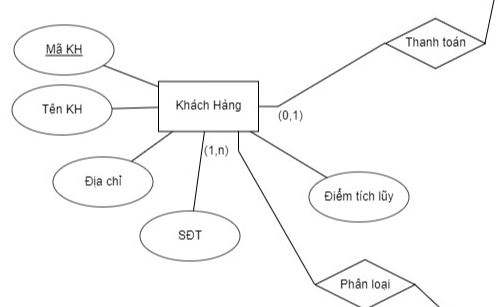
### 1.7 Bảng loại khác hàng (TYPECUSTOMER)



TYPECUSTOMER(idTypeCus, nameTypeCus,pointMarl,valueTypeCus)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idTypeCus | varchar(8) | Mã loại khách hàng | Khoá chính, not null |
| 2 | nameTypeCus | nvarchar(30) | Tên loại khách hàng | Not null |
| 3 | pointMark | int | Điểm tích lũy |  |
| 4 | valueTypeCus | int | Trị giá loại khách hàng |  |

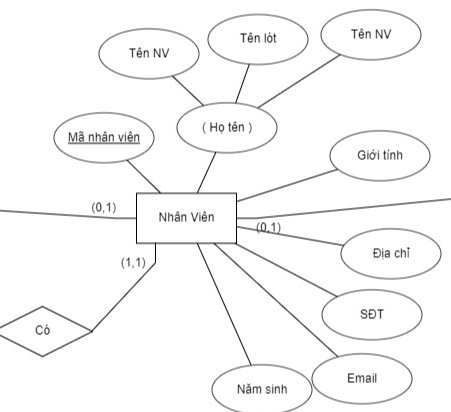
### 1.8 Bảng khách hàng (CUSTOMER)



CUSTOMER(idCus, nameCus, addressCus, phoneNumber, pointCus, idTypeCus).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idCus | varchar(8) | Mã khách hàng | Khoá chính, not null |
| 2 | nameCus | nvarchar(30) | Tên khách hàng | Not null |
| 3 | addressCus | nvarchar(30) | Địa chỉ khác hàng |  |
| 4 | phoneNumber | varchar(20) | Số điện thoại khách hàng |  |
| 5 | pointCus | int | Điểm tích lũy khách hàng | Not null |
| 6 | idTypeCus | varchar(8) | Mã loại khách hàng | Khóa ngoại |

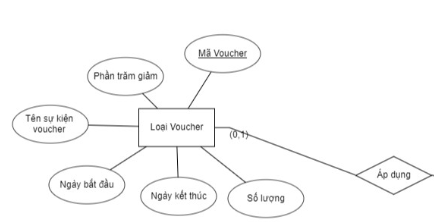
### 1.9 Bảng nhân viên (EMPLOYEE)



EMPLOYEE(idEmployee, firstName, middleName, lastName, sex, addEmp, phoneNumber, email, yearOfBirth)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idEmployee | varchar(8) | Mã nhân viên | Khoá chính, not null |
| 2 | firstName | nvarchar(10) | Họ nhân viên | not null |
| 3 | middleName | nvarchar(10) | Tên đệm nhân viên | not null |
| 4 | lastName | varchar(10) | Tên nhân viên | not null |
| 5 | sex | nvarchar(5) | Giới tính | not null |
| 6 | addEmp | nvarchar(30) | Địa chỉ nhân viên |  |
| 7 | phoneNumber | varchar(10) | Số điện thoại nhân viên |  |
| 8 | email | varchar(50) | Địa chỉ thư điện tử |  |
| 9 | yearOfBirth | date | Năm sinh NVbao gồm cả d/m/y của nhân viên |  |

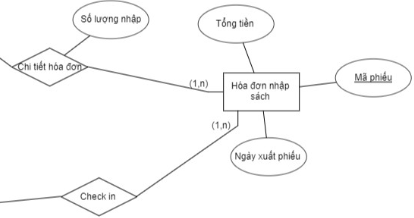
### 1.10 Bảng phiếu quà tặng (VOUCHER)



VOUCHER(idVoucher, valueVoucher, nameOfEventVoucher, dateStart, dateEnd, amount)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idVoucher | varchar(8) | Mã voucher | Khoá chính, not null |
| 2 | valueVoucher | int | Trị giá voucher | Not null |
| 3 | nameOfEventVoucher | nvarchar(40) | Tên sự kiện voucher |  |
| 4 | dateStart | date | Ngày bắt đầu voucher | Not null |
| 5 | dateEnd | date | Ngày kết thúc voucher | Not null |
| 6 | amount | int | Số voucher phát hành |  |

### 1.11 Bảng hóa đơn nhập sách (BILLINPUT)



BILLINPUT(idBillInput, dateOfInput, total, idEmployee)

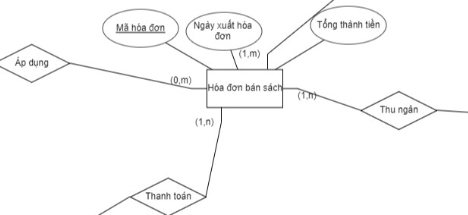
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBillInput | varchar(8) | Mã hóa đơn nhập sách | Khoá chính, not null |
| 2 | dateOfInput | date | Ngày nhập sách |  |
| 3 | total | int | Tổng tiền nhập | Not null |
| 4 | idEmployee | varchar(8) | Mã nhân viên | Khóa ngoại |

### 1.12 Bảng hóa đơn nhập theo sách (BOOK\_BILLINPUT)(bảng rã)

BOOK\_BILLINPUT(idBillInput, idBook, amountInput)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBillInput | varchar(8) | Mã hóa đơn nhập | Khoá chính, khóa ngoại, not null |
| 2 | idBook | varchar(8) | Mã sách | Khoá chính, khóa ngoại, not null |
| 3 | amountInput | int | Số lượng sách nhập | Not null |

### 1.13 Bảng hóa đơn bán sách (BILLOUTPUT)



BILLOUTPUT(idBillOutPut, dateOfBill, total ,idCus , idEmployee, idVoucher)

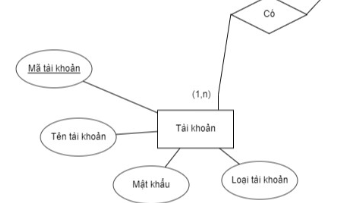
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBillOutPut | varchar(8) | Mã hóa đơn bán sách | Khoá chính, not null |
| 2 | dateOfBill | date | Ngày nhập hóa đơn |  |
| 3 | total | int | Tổng tiền bán sách | Not null |
| 4 | idCus | varchar(8) | Mã khách hàng | Khóa ngoại |
| 5 | idEmployee | varchar(8) | Mã nhân viên | Khóa ngoại |
| 6 | idVoucher | Varchar(8) | Mã voucher | Khóa ngoại |

### 1.14 Bảng hóa đơn bán theo sách (BOOK\_BILLOUTPUT)(bảng rã)

BOOK\_BILLOUTPUT(idBillOutput, idBook, amountOutput)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idBillOutput | varchar(8) | Mã hóa đơn bán sách | Khoá chính, khóa ngoại, not null |
| 2 | idBook | varchar(8) | Mã sách | Khoá chính, khóa ngoại, not null |
| 3 | amountOutput | int | Số lượng sách bán | Not null |

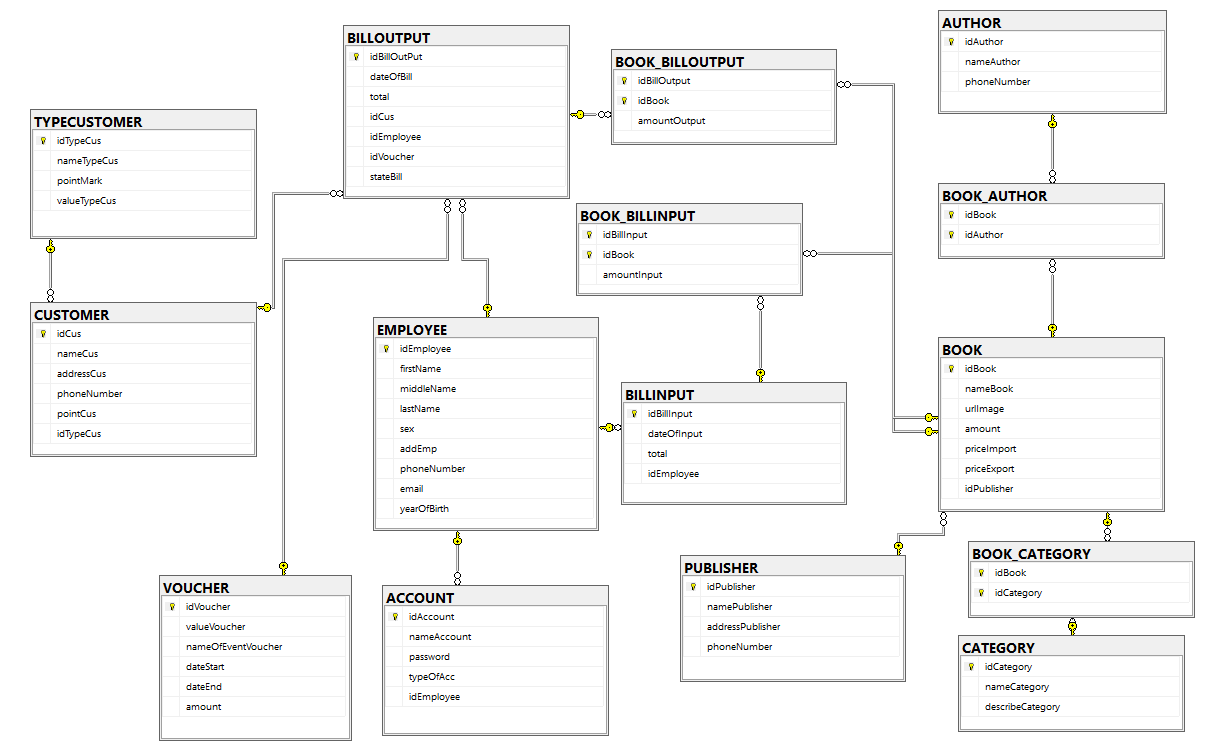
### 1.15 Bảng tài khoản (ACCOUNT)



ACCOUNT(idAccount, nameAccount, password, typeOfAcc, idEmployee)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| 1 | idAccount | varchar(8) | Mã tài khoản | Khoá chính, not null |
| 2 | nameAccount | varchar(20) | Tên đăng nhập tài khoản | Not null, unique |
| 3 | password | varchar(30) | Mật khẩu đăng nhập | Not null |
| 4 | typeOfAcc | bit | Phân quyền tài khoản | Not null |
| 5 | idEmployee | varchar(8) | Mã nhân viên | Khóa ngoại |

## \*\* Mô hình Diagram của đề tài:



## Mô tả khái quát các chức năng với các đối tượng

### **2.1 Đối với người quản lý**

Hỗ trợ người quản lý tìm kiếm các thông tin về các đầu sách, thông tin về hóa đơn, khách hàng, tình hình kho hàng sách, nhân viên…

Hỗ trợ đắc lực cho người quản lý giúp người quản lý nhận định một cách tống thể nhất về tình hình hoạt động, kinh doanh của cửa hàng mình

### 2.2 Đối với nhân viên

Hỗ trợ nhân viên tìm kiếm các thông tin liên quan đến sách như số lượng, đơn giá, khuyến mại giảm giá để nhân viên kịp thời có thông tin tiếp thị cho khách hàng, tìm kiếm các loại sách, tên sách một cách nhanh chóng, thống kê số lượng sách còn tồn trong kho.

Nhập số liệu một cách nhanh chóng, chính xác. Đảm bảo công việc lưu trữ chuẩn xác, tìm kiếm các thông tin về khách hàng, nhà cung cấp nhanh chóng, chính xác. Thống kê chính xác doanh thu của cửa hàng

### **2.3 Đối với khách hàng**

Xuất hóa đơn cho khách hàng mua hàng. Tìm kiếm thông tin sách cho khách hàng nếu khách hàng có yêu cầu tìm kiếm và kiểm tra xem số lượng sách khách hàng yêu cầu. Thay đổi thông tin khách hàng khi cần.

## Mô tả các chức năng chi tiết của hệ thống

### 3.1. Chức năng quản lý nhân viên

+ Cập nhật: Thêm, xóa, sửa thông tin nhân viên

+ Tìm kiếm nhân viên theo mã nhân viên, tên

+ Hiển thị danh sách các nhân viên các thông tin liên quan

### 3.2 Chức năng Quản lý khách hàng

+ Cập nhật: Thêm, xóa, sửa thông tin khách hàng thành viên

+ Tìm kiếm khách hàng thành viên theo số điện thoại, tên, mã

+ Hiển thị danh sách các khách hàng có các thông tin liên quan

### 3.3. Chức năng Quản lý kho sách

+ Cập nhật: Thêm, xóa, sửa thông tin sách, thông tin tác giả, thông tin nhà xuất bản, thể loại sách, giá tiền

+ Tìm kiếm sách theo tên, thể loại, tác giả

+ Hiển thị danh sách các quyển sách trong kho sách và thông tin liên quan

### 3.4 Chức năng Quản lý Voucher giảm giá

+ Cập nhật: Thêm, xóa, sửa thông tin các mã voucher

+ Tìm kiếm mã voucher theo tên, ngày

+ Hiển thị danh sách các mã giảm giá

### 3.5 Chức năng Quản lý hóa đơn

+ Tìm kiếm hóa đơn theo số điện thoại khách hàng, ngày mua

+ Hiển thị và xem chi tiết hóa đơn, các thông tin liên quan

### 3.6 Chức năng Quản lý phiếu nhập

+ Cập nhật: Thêm phiếu nhập, xóa, sửa hóa đơn nhập

+ Tìm kiếm phiếu nhập theo số điện thoại nhà xuất bản, ngày nhập

+ Hiển thị xem chi tiết phiếu nhập và xuất hóa đơn nhập

### 3.7. Chức năng Quản lý doanh thu

+ Quản lý và thống kê doanh thu

### 3.80 Chức năng lập hóa đơn bán hàng

+ Nhân viên sẽ thực hiện chức năng này khi có khách hàng đến mua hàng và mã nhân viên sẽ được lưu vào các hóa đơn

## Sơ lược về các ràng buộc

Nhân viên: Mỗi nhân viên phải có một id, quyền đăng nhập vào hệ thống, username, password và các thông tin như họ tên, địa chỉ, số điện thoại, lương, hình ảnh, ngày tạo, ngày update

+ Loại khách hàng: Mỗi loại khách hàng sẽ có một id, tên, mức ưu đãi, ngày tạo, ngày update

+ Khách hàng: Khi mua hàng lần đầu tiên phải đăng kí là thành viên của cửa hàng và mỗi thành viên sẽ có một id và các thông tin như học tên, địa chỉ, số điện thoại, loại thành viên, ngày tạo, ngày update và có điểm tích lũy

+ Sách: Mỗi quyển sách sẽ có một id và các thông tin như tên sách, bìa sách, thể loại, tên tác giả, nhà xuất bản, giá bán, số lượng sách trong kho, ngày tạo, ngày update

+ Voucher: Mỗi mã giảm giá có một id, ngày bắt đầu, ngày kết thúc, phần trăm giảm giá, tiêu đề giảm giá, ngày tạo, ngày update

+ Hóa đơn: Mỗi hóa đơn sẽ có một id và các thông tin như tên nhân viên, tên thành viên, ngày mua, mã giảm giá, giá trị hóa đơn, ngày tạo, ngày update

+ Chi tiết hóa đơn: Chi tiết hóa đơn sẽ có id hóa đơn, tên sách, số lượng, giá bán, ngày tạo, ngày update

+ Phiếu nhập: Mỗi phiếu nhập sẽ có một id và các thông tin như tên nhân viên, tên nhà xuất bản, ngày nhập, tổng tiền nhập, ngày tạo, ngày update

+ Chi tiết phiếu nhập: Chi tiết phiếu nhập sẽ có id phiếu nhập, tên sách, số lượng, giá nhập, ngày tạo, ngày update

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

* Áp dụng các ràng buộc toàn vẹn như: not null, unique, ràng buộc về khoá chính, khoá ngoại

\*\* Code SQL tạo cơ sở dữ liệu:

create database BOOKSTOREMANAGE

GOuse BOOKSTOREMANAGE

GOCREATE TABLE [dbo].AUTHOR (

idAuthor varchar(8) NOT NULL,

nameAuthor nvarchar(30) NOT NULL unique,

phoneNumber varchar(10) NULL

CONSTRAINT pk\_author PRIMARY KEY (idAuthor)

)

GO

CREATE TABLE [dbo].PUBLISHER (

idPublisher varchar(8) NOT NULL,

namePublisher nvarchar(30) NOT NULL,

addressPublisher nvarchar(30) NULL,

phoneNumber varchar(10) NULL

CONSTRAINT pk\_publisher PRIMARY KEY (idPublisher)

)

GO

CREATE TABLE [dbo].BOOK (

idBook varchar(8) NOT NULL,

nameBook nvarchar(20) NOT NULL,

urlImage image NULL,

amount int not null default 0,

priceImport int NULL,

priceExport int NULL,

idPublisher varchar(8) NULL

CONSTRAINT pk\_book PRIMARY KEY (idBook),

CONSTRAINT fk\_Publisher FOREIGN KEY (idPublisher) REFERENCES PUBLISHER (idPublisher),

CONSTRAINT chk\_amount CHECK (amount >= 0),

CONSTRAINT chk\_price CHECK (priceImport >= 0 and priceExport >= 0)

)

GO

CREATE TABLE [dbo].TYPECUSTOMER (

idTypeCus varchar(8) NOT NULL,

nameTypeCus nvarchar(30) NOT NULL,

pointMark int NULL,

valueTypeCus int NULL

CONSTRAINT pk\_typeCus PRIMARY KEY (idTypeCus),

CONSTRAINT chk\_pointMark CHECK (pointMark >= 0),

CONSTRAINT chk\_valueTypeCus CHECK (valueTypeCus >= 0 and valueTypeCus < 100)

)

GO

CREATE TABLE [dbo].CUSTOMER (

idCus varchar(8) NOT NULL,

nameCus nvarchar(30) NOT NULL,

addressCus nvarchar(30) NULL,

phoneNumber varchar(20) NULL,

pointCus int NOT NULL DEFAULT 0,

idTypeCus varchar(8)NULL

CONSTRAINT pk\_customer PRIMARY KEY (idCus),

CONSTRAINT fk\_TypeCus FOREIGN KEY (idTypeCus) REFERENCES TYPECUSTOMER (idTypeCus),

CONSTRAINT chk\_pointCus CHECK (pointCus >= 0)

)

GO

CREATE TABLE [dbo].EMPLOYEE (

idEmployee varchar(8) NOT NULL,

firstName nvarchar(10) NOT NULL,

middleName nvarchar(10) NULL,

lastName varchar(10) NOT NULL,

sex nvarchar(5) NOT NULL,

addEmp nvarchar(30) NULL,

phoneNumber varchar(10) NULL,

email varchar(50) NOT NULL,

yearOfBirth date NULL,

CONSTRAINT chk\_sex CHECK (upper(sex) like N'NAM' or upper(sex) like N'NỮ' or upper(sex) like N'KHÁC'),

CONSTRAINT chk\_email CHECK (email like '%@gmail.com%'),

CONSTRAINT chk\_yearOfBirth CHECK (DATEDIFF(YEAR,yearOfBirth,GETDATE()) >= 18 AND DATEDIFF(YEAR,yearOfBirth,GETDATE()) <= 50),

CONSTRAINT pk\_employee PRIMARY KEY (idEmployee)

)

GO

CREATE TABLE [dbo].VOUCHER (

idVoucher varchar(8) NOT NULL,

valueVoucher int NOT NULL,

nameOfEventVoucher nvarchar(40) NULL,

dateStart date not null,

dateEnd date not null,

amount int NULL,

CONSTRAINT pk\_voucher PRIMARY KEY (idVoucher),

CONSTRAINT chk\_amountVoucher CHECK (amount>=0)

)

GO

CREATE TABLE [dbo].BILLOUTPUT (

idBillOutPut varchar(8) NOT NULL,

dateOfBill date NULL,

total int NOT NULL DEFAULT 0,

idCus varchar(8)NULL,

idEmployee varchar(8) NULL,

idVoucher varchar(8) null,

stateBill bit default 0,

CONSTRAINT pk\_billOutPut PRIMARY KEY (idBillOutPut),

CONSTRAINT fk\_pay FOREIGN KEY (idCus) REFERENCES CUSTOMER (idCus),

CONSTRAINT fk\_collect FOREIGN KEY (idEmployee) REFERENCES EMPLOYEE (idEmployee),

CONSTRAINT fk\_Discount FOREIGN KEY (idVoucher) REFERENCES VOUCHER (idVoucher),

)

GO

CREATE TABLE [dbo].ACCOUNT (

idAccount varchar(8) NOT NULL,

nameAccount varchar(20) NOT NULL unique,

password varchar(30) NOT NULL,

typeOfAcc bit NOT NULL,

idEmployee varchar(8) NULL

CONSTRAINT pk\_account PRIMARY KEY (idAccount),

CONSTRAINT fk\_acc\_Of\_Employee FOREIGN KEY (idEmployee) REFERENCES EMPLOYEE (idEmployee),

CONSTRAINT chk\_userName CHECK (len(nameAccount) > 6),

CONSTRAINT chk\_passwd CHECK (len(password) > 6)

)

GO

CREATE TABLE [dbo].CATEGORY (

idCategory varchar(8) NOT NULL,

nameCategory nvarchar(20) NOT NULL unique,

describeCategory nvarchar(50) NULL,

CONSTRAINT pk\_category PRIMARY KEY (idCategory)

)

GO

CREATE TABLE [dbo].BOOK\_AUTHOR (

idBook varchar(8) NOT NULL,

idAuthor varchar(8) NOT NULL,

CONSTRAINT pk\_book\_author PRIMARY KEY (idBook,idAuthor),

CONSTRAINT fk\_book FOREIGN KEY (idBook) REFERENCES BOOK (idBook),

CONSTRAINT fk\_author FOREIGN KEY (idAuthor) REFERENCES AUTHOR (idAuthor)

)

GO

CREATE TABLE [dbo].BOOK\_CATEGORY (

idBook varchar(8) NOT NULL,

idCategory varchar(8) NOT NULL,

CONSTRAINT pk\_book\_category PRIMARY KEY (idBook,idCategory),

CONSTRAINT fk\_id\_into\_book FOREIGN KEY (idBook) REFERENCES BOOK (idBook),

CONSTRAINT fk\_id\_into\_categgory FOREIGN KEY (idCategory) REFERENCES CATEGORY (idCategory)

)

GO

CREATE TABLE [dbo].BILLINPUT (

idBillInput varchar(8) NOT NULL,

dateOfInput date NULL,

total int NOT NULL DEFAULT 0,

idEmployee varchar(8) NULL,

CONSTRAINT pk\_billinput PRIMARY KEY (idBillInput),

CONSTRAINT fk\_employee\_checkin FOREIGN KEY (idEmployee) REFERENCES EMPLOYEE (idEmployee)

)

GO

CREATE TABLE [dbo].BOOK\_BILLINPUT (

idBillInput varchar(8) NOT NULL,

idBook varchar(8) NOT NULL,

amountInput INT NOT NULL,

CONSTRAINT pk\_book\_billinput PRIMARY KEY (idBillInput,idBook),

CONSTRAINT fk\_idBillInput FOREIGN KEY (idBillInput) REFERENCES BILLINPUT (idBillInput),

CONSTRAINT fk\_idBook\_into\_input FOREIGN KEY (idBook) REFERENCES BOOK (idBook),

CONSTRAINT chk\_amountInput CHECK(amountInput>=0)

)

GOCREATE TABLE [dbo].BOOK\_BILLOUTPUT (

idBillOutput varchar(8) NOT NULL,

idBook varchar(8) NOT NULL,

amountOutput int NOT NULL

CONSTRAINT pk\_book\_billoutput PRIMARY KEY (idBillOutput,idBook),

CONSTRAINT fk\_idBillOuput FOREIGN KEY (idBillOutput) REFERENCES BILLOUTPUT (idBillOutput),

CONSTRAINT fk\_idBook\_into\_output FOREIGN KEY (idBook) REFERENCES BOOK (idBook),

CONSTRAINT chk\_amountOutput CHECK(amountOutput>=0)

)

GO

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG,CÁC CHỨC NĂNG

*Note: viết bắng thủ tục, hàm của SQL có sử dụng view, trigger*

## \*\*\*Code kết nối cơ sở dữ liệu:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

using System.Windows.Forms;

namespace Proj\_Book\_Store\_Manage.DBLayer

{

public class DBMain

{

public static string ServerName = "TANDAT";

public static string DatabaseName = "BOOKSTOREMANAGE";

public static string UserName= "";

public static string Password= "";

private string ConnStr = $@"Data Source={ServerName}; Initial Catalog={DatabaseName};User Id={UserName};Password={Password}";

private SqlConnection conn = null;

private SqlCommand cmd = null;

private SqlDataAdapter adapter = null;

private DataTable dt = null;

public DBMain()

{

conn = new SqlConnection(ConnStr);

cmd = conn.CreateCommand();

dt = new DataTable();

}

//Trả về Datatable để load dữ liệu lên form từ view

public DataTable LoadData(string nameView, CommandType ct)

{

dt = new DataTable();

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

conn.Open();

string cmm = "select \* from "+ nameView;

adapter = new SqlDataAdapter(cmm,conn);

adapter.Fill(dt);

conn.Close();

return dt;

}

//Trả về Datatable để thực thi function

public DataTable ExecuteFunction(SqlCommand cmdFunction, ref string error)

{

error = "";

dt = new DataTable();

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

cmd = cmdFunction;

cmd.Connection = conn;

try

{

adapter = new SqlDataAdapter(cmd);

adapter.Fill(dt);

}

catch (SqlException ex)

{

error = ex.Message;

MessageBox.Show(error);

dt = null;

}

finally

{

conn.Close();

}

return dt;

}

//dư -- có thể xóa

public string ExecuteFunctionToString(SqlCommand cmdFunction, ref string error)

{

string valueReturn = null;

error = "";

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

cmd = cmdFunction;

cmd.Connection = conn;

try

{

valueReturn = cmd.ExecuteScalar().ToString();

}

catch (SqlException ex)

{

error = ex.Message;

MessageBox.Show(error);

}

finally

{

conn.Close();

}

return valueReturn;

}

public int ExecuteFunctionToInt(SqlCommand cmdFunction, ref string error)

{

int valueReturn = -1;

error = "";

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

cmd = cmdFunction;

cmd.Connection = conn;

try

{

valueReturn = int.Parse(cmd.ExecuteScalar().ToString());

}

catch (SqlException ex)

{

error = ex.Message;

MessageBox.Show(error);

}

finally

{

conn.Close();

}

return valueReturn;

}

//Trả về true, false thực thi procedure

public bool ExecuteProcedure(string sqlProcedure, CommandType ct, List<SqlParameter> parameters, ref string error)

{

bool f = false;

error = "";

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

cmd.CommandText = sqlProcedure;

cmd.CommandType = ct;

cmd.Parameters.Clear();

if (parameters != null)

{

foreach (SqlParameter i in parameters)

{

cmd.Parameters.Add(i);

}

}

try

{

cmd.ExecuteNonQuery();

f = true;

}

catch (SqlException ex)

{

error = ex.Message;

}

finally

{

conn.Close();

}

return f;

}

// Trả về table thực thi procedure

public DataTable Procedure(string sqlProcedure, CommandType ct, List<SqlParameter> parameters, ref string error)

{

error = "";

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

cmd.CommandText = sqlProcedure;

cmd.CommandType = ct;

cmd.Parameters.Clear();

foreach (SqlParameter i in parameters)

{

cmd.Parameters.Add(i);

}

try

{

adapter = new SqlDataAdapter(cmd);

dt = new DataTable();

adapter.Fill(dt);

}

catch (SqlException ex)

{

error = ex.Message;

}

finally

{

conn.Close();

}

return dt;

}

}

}

## Danh sách các trigger của đề tài

### 1.1 Kiểm tra số lượng sách thêm trong giỏ hàng phải luôn có đủ trong kho

--Kiểm tra số lượng sách thêm trong giỏ hàng phải luôn có đủ trong kho

create or alter trigger trg\_checkInsertCart

on dbo.BOOK\_BILLOUTPUT

for insert

as

begin

declare @numBookSold int, @numBookRemain int, @idBook varchar(8)

select @numBookSold=i.amountOutput, @idBook=i.idBook

from inserted i

set @numBookRemain=(select dbo.BOOK.amount from dbo.BOOK where dbo.BOOK.idBook=@idBook)

if (@numBookSold > @numBookRemain)

begin

raiserror('Số lượng sách trong kho không đáp ứng đủ !',16,1)

rollback transaction

end

end

go

### 1.2 Kiểm tra số lượng sách trong kho có đủ để thêm vào BOOK\_BILLOUTPUT

create or alter trigger trg\_checkUpdateCart

on dbo.BOOK\_BILLOUTPUT

for update

as

begin

declare @numBookSold int, @numBookRemain int, @numBookDelete int, @idBookInsert varchar(8), @idBookDelete varchar(8)

select @numBookSold=i.amountOutput,@numBookDelete=d.amountOutput, @idBookInsert=i.idBook,@idBookDelete=d.idBook

from inserted i, deleted d

if(charindex(@idBookInsert,@idBookDelete)>0)

set @numBookRemain=((select BOOK.amount from dbo.BOOK where @idBookInsert=BOOK.idBook)+@numBookDelete)

else

set @numBookRemain=(select BOOK.amount from dbo.BOOK where @idBookInsert=BOOK.idBook)

if (@numBookSold > @numBookRemain)

begin

raiserror('Số lượng sách trong kho không đáp ứng đủ !',16,1)

rollback transaction

end

end

go

### 1.3 Kiểm tra điều kiện voucher áp dụng vào đơn hàng

create or alter trigger trg\_discountAndUpdateVoucher

on dbo.BILLOUTPUT

for update

as

begin

declare @idBillDiscount varchar(8), @idVcher varchar(8), @totalAfterDiscount int, @totalBeforeDiscount int, @checkBill bit, @amountVoucher int

select @idBillDiscount=i.idBillOutPut, @idVcher=i.idVoucher, @totalBeforeDiscount=d.total, @totalAfterDiscount=i.total

from inserted i, deleted d

--Kiểm tra idVoucher có tồn tại

if (@idVcher is not null and not exists(select \* from dbo.VOUCHER where @idVcher=dbo.VOUCHER.idVoucher))

begin

raiserror ('Voucher không tồn tại trong kho',16,1)

rollback transaction

end

if(@idVcher is not null)

begin

--Kiểm tra nếu thêm voucher thì có vượt quá 50% giá trị bill

if (@totalAfterDiscount <= (0.5\*@totalBeforeDiscount))

begin

raiserror ('Voucher không thể áp dụng cho đơn hàng này',16,1)

rollback transaction

end

set @amountVoucher=(select dbo.VOUCHER.amount from dbo.VOUCHER where dbo.VOUCHER.idVoucher=@idVcher)

--Kiểm tra số lượng voucher có còn còn trong kho

if(@amountVoucher<=0)

begin

raiserror ('Voucher không đủ số lượng trong kho',16,1)

rollback transaction

end

--Kiểm tra Voucher có còn hạn sử dụng

select @checkBill=dbo.func\_checkVoucherValidOrNot(VOUCHER.dateStart,VOUCHER.dateEnd,BILLOUTPUT.dateOfBill)

from dbo.BILLOUTPUT, dbo.VOUCHER

where BILLOUTPUT.idBillOutPut=@idBillDiscount and VOUCHER.idVoucher=@idVcher

if (@checkBill=0)

begin

raiserror ('Voucher đã hết hạn sử dụng',16,1)

rollback transaction

end

end

end

go

### 1.4 Kiểm tra điều kiện thỏa ràng buộc số điện thoại trong bảng AUTHOR

create or alter trigger trg\_checkPhoneNumberOfAuthor

on AUTHOR

for insert, update

as

begin

declare @phoneNumber varchar(20)

select @phoneNumber=i.phoneNumber

from inserted i

if (dbo.func\_checkPhone(@phoneNumber)=0)

begin

raiserror ('Số điện thoại nhập không đúng định dạng',16,1)

rollback transaction

end

end

go

### 1.5 Kiểm tra điều kiện thỏa ràng buộc số điện thoại trong bảng PUBLISHER

create or alter trigger trg\_checkPhoneNumberOfPublisher on PUBLISHER

for insert, update

as

begin

declare @phoneNumber varchar(20)

select @phoneNumber=i.phoneNumber

from inserted i

if (dbo.func\_checkPhone(@phoneNumber)=0)

begin

raiserror ('Số điện thoại nhập không đúng định dạng',16,1)

rollback transaction

end

End

go

g

### 1.6 Kiểm tra số điện thoại của khách hàng

create or alter trigger trg\_checkPhoneNumberOfCustomer

on CUSTOMER

for insert, update

as

begin

declare @phoneNumber varchar(20)

select @phoneNumber=i.phoneNumber

from inserted i

if (dbo.func\_checkPhone(@phoneNumber)=0)

begin

raiserror ('Số điện thoại nhập không đúng định dạng',16,1)

rollback transaction

end

end

go

### 1.7 Kiểm tra giá nhập sách phải bé hơn giá bán ra trong bảng BOOK

create or alter trigger trg\_checkPriceOfBook

on BOOK

for insert, update

as

begin

declare @priceImport int, @priceExport int

select @priceImport=i.priceImport, @priceExport=i.priceExport

from inserted i

if (@priceImport > @priceExport)

begin

raiserror ('Giá nhập sách phải nhỏ hơn giá bán',16,1)

rollback transaction

end

end

go

### 1.8 Kiểm tra phần trăm được giảm của khách hàng không lớn hơn 100% và không bé hơn 0%

Create or alter trigger trg\_checkTypeCustomer on TYPECUSTOMER

for insert, update

as

begin

declare @value int

select @value = i.valueTypeCus from inserted i

if (@value < 0 or @value >100)

begin

print N'Nhập giá trị value không chính xác. Value nằm trong khoảng từ 0 -> 100'

rollback transaction

end

end

go

### 1.9 Kiểm tra ngày bắt đầu phải nhỏ hơn hoặc bằng ngày kết thúc của VOUCHER

Create or alter trigger trg\_checkDateVoucher on Voucher

for insert, update

as

begin

declare @dateStart date, @dateEnd date

select @dateStart = i.dateStart, @dateEnd = i.dateEnd

from inserted i

if (@dateStart > @dateEnd)

begin

print N'Ngày bắt đầu và kết thúc không hợp lệ'

rollback transaction

end

end

Go

## Danh sách các Function của đề tài

### 2.1 Hàm func\_checkVoucherValidOrNot

- Hàm trả về 1 nếu voucher còn thời hạn sử dụng, 0 nếu đã hết thời hạn sử dụng

create or alter function func\_checkVoucherValidOrNot(@dateStart date, @dateEnd date, @dateCurrent date)

returns bit

as

begin

declare @dayOfVoucher int, @dayCurrentSinceVoucher int

set @dayOfVoucher=DATEDIFF(day,@dateEnd,@dateStart)

set @dayCurrentSinceVoucher=DATEDIFF(day,@dateCurrent,@dateStart)

if (@dayCurrentSinceVoucher < @dayOfVoucher)

return 0

return 1

end

go

### 2.2 Hàm func\_checkPhone

- Hàm kiểm tra định dạng của số điện thoại. Trả về 0 nếu sai định dạng, trả về 1 nếu đúng định dạng

create or alter function func\_checkPhone(@phoneNumber varchar(20))

returns bit

as

begin

declare @firstChar varchar(2)

set @firstChar=LEFT(@phoneNumber,1)

if (ISNUMERIC(@phoneNumber)=0 or len(@phoneNumber)!=10 or @firstChar not like '0')

return 0

return 1

end

go

### 2.3. Hàm func\_getAllCategoryOfBook

- Hàm trả về danh sách các thể loại của sách

create or alter function func\_getAllCategoryOfBook(@idBook varchar(8) )

returns table

as

return (select distinct dbo.CATEGORY.nameCategory as TheLoai

from dbo.CATEGORY, dbo.BOOK\_CATEGORY

where dbo.BOOK\_CATEGORY.idBook=@idBook and dbo.CATEGORY.idCategory=dbo.BOOK\_CATEGORY.idCategory)

go

### 2.4 Hàm func\_getAllAuthorOfBook

- Trả về danh sách tác giả của sách

create or alter function func\_getAllAuthorOfBook(@idBook varchar(8) )

returns table

as

return (select distinct dbo.AUTHOR.nameAuthor as TacGia

from dbo.AUTHOR, dbo.BOOK\_AUTHOR

where dbo.BOOK\_AUTHOR.idBook=@idBook and dbo.AUTHOR.idAuthor=dbo.BOOK\_AUTHOR.idAuthor)

go

### 2.5 Hàm func\_getIDCategory

- Hàm trả về ID của thể loại

create or alter function func\_getIDCategory(@nameCategory nvarchar(20))

returns varchar(8)

as

begin

declare @idCategory varchar(8)

select @idCategory=CATEGORY.idCategory

from dbo.CATEGORY

where CATEGORY.nameCategory=@nameCategory

return @idCategory

end

go

### 2.6. Hàm func\_getIDAuthor

- Hàm trả về ID của tác giả, tham số là tên tác giả

create or alter function func\_getIDAuthor(@nameAuthor nvarchar(30))

returns varchar(8)

as

begin

declare @idAuthor varchar(8)

select @idAuthor=AUTHOR.idAuthor

from dbo.AUTHOR

where AUTHOR.nameAuthor=@nameAuthor

return @idAuthor

end

go

### 2.7. Hàm func\_returnIdBillFalse

- Hàm trả về ID hóa đơn bán sách nếu trạng thái hóa đơn chưa thanh toán

create or alter function func\_returnIdBillFalse ()

returns varchar(8)

as

begin

declare @idBill varchar (8)

select @idBill = idBillOutPut from BILLOUTPUT

where stateBill = 0

return @idBill

end

go

### 2.8. Hàm func\_checkBookIsInBill

- Hàm kiểm tra xem ID sách này có trong đơn hàng chưa. Trả về 1 nếu đã tồn tại trong đơn hàng, trả về 0 nếu chưa tồn tại.

create or alter function func\_checkBookIsInBill(@idBill varchar(8) , @idBook varchar(8) )

returns bit

as

begin

if exists(select \*

from dbo.BOOK\_BILLOUTPUT

where BOOK\_BILLOUTPUT.idBillOutput=@idBill and BOOK\_BILLOUTPUT.idBook=@idBook)

return 1

return 0

end

go

### 2.9. Hàm func\_getDataOfBillExport

- Hàm trả về thông tin chi tiết hóa đơn bán hàng

create or alter function func\_getDataOfBillExport(@idBill varchar(8) )

returns table

as

return ( select BOOK\_BILLOUTPUT.idBook as N'ID', BOOK.nameBook as N'Tên Sách', BOOK\_BILLOUTPUT.amountOutput as N'Số lượng'

from dbo.BOOK\_BILLOUTPUT, dbo.BOOK

where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBill and dbo.BOOK\_BILLOUTPUT.idBook=BOOK.idBook )

go

### 2.10. Hàm func\_getDataOfBillImport

- Hàm trả về thông tin chi tiết của hóa đơn nhập sách

create or alter function func\_getDataOfBillImport(@idBill varchar(8) )

returns table

as

return ( select BOOK\_BILLINPUT.idBook as N'ID', BOOK.nameBook as N'Tên Sách', BOOK\_BILLINPUT.amountInput as N'Số lượng'

from dbo.BOOK\_BILLINPUT, dbo.BOOK

where dbo.BOOK\_BILLINPUT.idBillInput=@idBill and dbo.BOOK\_BILLINPUT.idBook=BOOK.idBook )

go

### 2.11. Hàm func\_getNameCusOfBillOutPut

- Hàm trả về tên của khách hàng với tham số là mã hóa đơn

create or alter function func\_getNameCusOfBillOutPut(@idBill varchar(8) )

returns varchar(30)

begin

return (select dbo.CUSTOMER.nameCus

from dbo.CUSTOMER, dbo.BILLOUTPUT

where dbo.CUSTOMER.idCus=dbo.BILLOUTPUT.idCus and dbo.BILLOUTPUT.idBillOutPut=@idBill)

end

go

### 2.12. Hàm func\_getNameEmpOfBillOutPut

- Hàm trả về tên nhân viên với tham số là mã hóa đơn

create or alter function func\_getNameEmpOfBillOutPut(@idBill varchar(8) )

returns varchar(30)

begin

return (select CONCAT(dbo.EMPLOYEE.firstName,dbo.EMPLOYEE.middleName,dbo.EMPLOYEE.lastName)

from dbo.EMPLOYEE, dbo.BILLOUTPUT

where dbo.EMPLOYEE.idEmployee=dbo.BILLOUTPUT.idEmployee and dbo.BILLOUTPUT.idBillOutPut=@idBill)

end

go

### 2.13. Hàm func\_getNameBookById

- Hàm trả về tên sách với tham số là mã sách

create or alter function func\_getNameBookById(@idBook varchar(8) )

returns varchar(20)

begin

return (select dbo.BOOK.nameBook

from dbo.BOOK

where dbo.BOOK.idBook=@idBook)

end

go

### 2.14. Hàm func\_getAmountBookById

- Hàm trả về số lượng sách trong kho với tham số là mã sách

create or alter function func\_getAmountBookById(@idBook varchar(8) )

returns int

begin

return (select dbo.BOOK.amount

from dbo.BOOK

where dbo.BOOK.idBook=@idBook)

end

Go

### 2.15. Hàm func\_returnToTalOfBillOutput

- Hàm trả về tổng số tiền khách hàng phải trả khi xuất hóa đơn.

create or alter function func\_returnToTalOfBillOutput(@idBill varchar(8) )

returns int

begin

if not exists(select \* from dbo.BOOK\_BILLOUTPUT

where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBill)

return 0

return (select sum(Q.TotalBook)

from (select dbo.BOOK\_BILLOUTPUT.amountOutput\*dbo.BOOK.priceExport as TotalBook from dbo.BOOK\_BILLOUTPUT, dbo.BOOK where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBill and dbo.BOOK.idBook=dbo.BOOK\_BILLOUTPUT.idBook) as Q)

En d

go

### 2.16. Hàm func\_returnToTalOfBillInput

- Hàm trả về tổng số tiền khi nhập sách.

create or alter function func\_returnToTalOfBillInput(@idBill varchar(8) )

returns int

begin

if not exists(select \* from dbo.BOOK\_BILLINPUT

where dbo.BOOK\_BILLINPUT.idBillInput=@idBill)

return 0

return (select sum(Q.TotalBook)

from (select dbo.BOOK\_BILLINPUT.amountInput\*dbo.BOOK.priceImport as TotalBook from dbo.BOOK\_BILLINPUT, dbo.BOOK where dbo.BOOK\_BILLINPUT.idBillInput=@idBill and dbo.BOOK.idBook=dbo.BOOK\_BILLINPUT.idBook) as Q)

end

go

### 2.17. Hàm func\_getIdEmployee

- Hàm trả về mã nhân viên, tên nhân viên khi đăng nhập.

create or alter function func\_getIdEmployee (@user varchar(20), @password varchar(30))

returns @table table (idEmp varchar(8) ,nameEmp nvarchar (40),idRole bit)

as

begin

declare @idEmp varchar(8) , @firstName nvarchar(10), @middleName nvarchar(10), @lastName nvarchar (10), @name nvarchar (40), @idRole bit

select @idEmp = ACCOUNT.idEmployee,

@middleName = middleName,

@firstName = firstName,

@lastName = lastName,

@idRole = typeOfAcc

from ACCOUNT, EMPLOYEE

where nameAccount=@user and

password = @password and

ACCOUNT.idEmployee = EMPLOYEE.idEmployee

set @name = @firstName + ' ' + @middleName + ' ' + @lastName

insert into @table values (@idEmp, @name, @idRole)

return

end

go

### 2.18. Hàm func\_getValueDiscountOfTypeCustomer

- Hàm trả về giá trị được giảm giá của khách hàng.

create or alter function func\_getValueDiscountOfTypeCustomer(@idCustomer varchar(8) )

returns int

as

begin

declare @discountOfTypeCus int

select @discountOfTypeCus=dbo.TYPECUSTOMER.valueTypeCus

from dbo.TYPECUSTOMER, dbo.CUSTOMER

where dbo.CUSTOMER.idCus=@idCustomer and dbo.CUSTOMER.idTypeCus=dbo.TYPECUSTOMER.idTypeCus

return @discountOfTypeCus

end

go

### 2.19. Hàm func\_searchAccount

- Hàm trả về thông tin của tài khoản

Create or alter function func\_searchAccount (@idAccount varchar(8), @username varchar(20))

returns table

as

return (select \* from ACCOUNT

where nameAccount = @username or

idAccount = @idAccount)

go

### 2.20 Hàm func\_searchAuthor

- Hàm trả về thông tin tác giả.

Create or alter function func\_searchAuthor (@idAuthor varchar(8) , @nameAuthor nvarchar(30))

returns table

as

return (

select \* from AUTHOR

where idAuthor = @idAuthor or

nameAuthor = @nameAuthor)

go

### 2.21. Hàm func\_searchCategory

- Hàm trả về thông tin của thể loại sách.

create or alter function func\_searchCategory (@idCategory varchar(8) , @nameCategory nvarchar(20))

returns table

as

return (

select \* from CATEGORY

where idCategory = @idCategory or

nameCategory = @nameCategory)

go

### 2.22. Hàm func\_searchEmployee

- Hàm trả về thông tin của nhân viên.

Create or alter function func\_searchEmployee (@idEmployee varchar(8) , @nameEmployee nvarchar (40))

returns table

as

return (

select idEmployee, firstName, middleName, lastName, sex, addEmp, phoneNumber, email, yearOfBirth

from EMPLOYEE

where idEmployee = @idEmployee or

(firstName + ' ' + middleName + ' '+lastName = @nameEmployee))

go

### 2.23. Hàm func\_searchCustomer

- Hàm trả về thông tin của khách hàng.

Create or alter function func\_searchCustomer (@idCustomer varchar(8) , @nameCustomer nvarchar(40))

returns table

as

return (

select \* from CUSTOMER

where idCus =@idCustomer or

nameCus = @nameCustomer)

go

### 2.24. Hàm func\_searchPublisher

- Hàm trả về thông tin nhà xuất bản.

Create or alter function func\_searchPublisher (@idPublisher varchar(8) , @namePublisher nvarchar(80))

returns table

as

return (

select \* from PUBLISHER

where idPublisher =@idPublisher or

namePublisher = @namePublisher)

go

### 2.25. Hàm func\_searchReceiptExport

- Hàm trả về thông tin hóa đơn bán sách.

Create or alter function func\_searchReceiptExport (@idBill varchar(8) )

returns table

as

return (

select \* from BILLOUTPUT where idBillOutPut = @idBill)

go

### 2.26. Hàm func\_searchVoucher

- Hàm trả về thông tin của Voucher.

Create or alter function func\_searchVoucher(@idVoucher varchar(8) )

returns table

as

return (

select \* from VOUCHER where idVoucher = @idVoucher)

go

### 2.27. Hàm func\_searchTypeCustomer

- Hàm trả về thông tin của loại khách hàng.

Create or alter function func\_searchTypeCustomer(@id varchar(8) , @name nvarchar(30))

returns table

as

return (

select \* from TYPECUSTOMER

where idTypeCus = @id or nameTypeCus = @name

)

Go

### 2.28. Hàm func\_searchReceiptImport

- Hàm trả về thông tin hóa đơn nhập sách

Create or alter function func\_searchReceiptImport(@idBill varchar(8) )

returns table

as

return (

select \* from BILLINPUT where idBillInput = @idBill)

go

### 2.29. Hàm func\_searchBook

- Hàm trả về thông tin sách

Create or alter function func\_searchBook (@idBook varchar(8), @nameBook nvarchar(20), @nameCategory nvarchar (20), @nameAuhtor nvarchar (30))

returns table

as

return (

select BOOK.idBook, BOOK.nameBook, BOOK.urlImage, BOOK.amount, BOOK.priceImport, BOOK.priceExport, book.idPublisher from BOOK inner join (

select distinct (BOOK.idBook)

from AUTHOR, BOOK\_AUTHOR, BOOK, CATEGORY, BOOK\_CATEGORY

where (AUTHOR.nameAuthor like @nameAuhtor and AUTHOR.idAuthor = BOOK\_AUTHOR.idAuthor and BOOK\_AUTHOR.idBook = BOOK.idBook) or

(CATEGORY.nameCategory like @nameCategory and CATEGORY.idCategory = BOOK\_CATEGORY.idCategory and BOOK\_CATEGORY.idBook = BOOK.idBook) or

(BOOK.idBook = @idBook or BOOk.nameBook = @nameBook)

)A on BOOK.idBook = A.idBook)

Go

## Danh sách các Procedure của đề tài

### Procedure thêm dữ liệu vào các bản

#### Procedure thêm một tài khoản mới

Chức năng tạo tài khoản chỉ dành cho admin và tài khoản được tạo ra sẽ có quyền admin hoặc nhân viên tùy vào lựa chọn

create or alter procedure proc\_addNewAccount (@idAccount varchar(8) , @nameAccount varchar(20), @password varchar(30), @typeOfAcc bit, @idEmployee varchar(8))as

begin

begin transaction addNewAcc

begin try

insert into dbo.ACCOUNT

(

idAccount,

nameAccount,

password,

typeOfAcc,

idEmployee

)

values

(

@idAccount,

@nameAccount,

@password,

@typeOfAcc,

@idEmployee

)

--tạo quyền cho user vừa thêm

exec proc\_createUser @nameAccount, @password

commit transaction addNewAcc

end try

begin catch

print (error\_message())

rollback transaction addNewAcc

end catch

end

#### Procedure thêm một tác giả mới vào dữ liệu

Tác giả được thêm vào sẽ có các thông tin như mã ID, tên tác giả và số điện thoại

create or alter procedure proc\_addNewAuthor

@idAuthor varchar(8) ,

@nameAuthor nvarchar(30),

@phoneNumber varchar(10)

as

begin

insert into dbo.AUTHOR

(

idAuthor,

nameAuthor,

phoneNumber

)

values

(

@idAuthor,

@nameAuthor,

@phoneNumber

)

end

#### Procedure thêm một sách mới vào dữ liệu

create or alter procedure proc\_addNewBook

@idBook varchar(8) ,

@nameBook nvarchar(20),

@urlImage image,

@priceImport int,

@priceExport int,

@idPublisher varchar(8)

as

begin

insert into dbo.BOOK

(

idBook,

nameBook,

urlImage,

priceImport,

priceExport,

idPublisher

)

values

(

@idBook,

@nameBook,

@urlImage,

@priceImport,

@priceExport,

@idPublisher

)

End

#### Procedure thêm một tác giả mới sách

create or alter procedure proc\_addNewBookAuthor

@idBook varchar(8) ,

@nameAuthor nvarchar(30)

as

begin

declare @idAuthor varchar(8)

select @idAuthor=dbo.func\_getIDAuthor(@nameAuthor)

insert into dbo.BOOK\_AUTHOR

(

idBook,

idAuthor

)

values

(

@idBook,

@idAuthor

)

End

#### Procedure thêm một thể loại mới cho sách

create or alter procedure proc\_addNewBookCategory

@idBook varchar(8) ,

@nameCategory nvarchar(20)

as

begin

declare @idCategory varchar(8)

select @idCategory=dbo.func\_getIDCategory(@nameCategory)

insert into dbo.BOOK\_CATEGORY

(

idBook,

idCategory

)

values

(

@idBook,

@idCategory

)

End

#### Procedure thêm một thể loại mới vào dữ liệu

create or alter procedure proc\_addNewCategory

@idCategory varchar(8) ,

@nameCategory nvarchar(20),

@describleCategory nvarchar(50)

as

begin

insert into dbo.CATEGORY

(

idCategory,

nameCategory,

describeCategory

)

values

(

@idCategory,

@nameCategory,

@describleCategory

)

End

#### Procedure thêm một thông tin nhân viên mới

create or alter procedure proc\_addNewEmployee

@idEmployee varchar(8) ,

@firstName nvarchar(10),

@middleName nvarchar(10),

@lastName nvarchar(10),

@sex nvarchar(5),

@addEmp nvarchar(30),

@phoneNumber varchar(10),

@email varchar(50),

@yearOfBirth date

as

begin

insert into dbo.EMPLOYEE

(

idEmployee,

firstName,

middleName,

lastName,

sex,

addEmp,

phoneNumber,

email,

yearOfBirth

)

values

(

@idEmployee,

@firstName,

@middleName,

@lastName,

@sex,

@addEmp,

@phoneNumber,

@email,

@yearOfBirth

)

end

#### Procedure thêm thông tin nhà xuất bản mới

create or alter procedure proc\_addNewPublisher

@idPublisher varchar(8) ,

@namePublisher nvarchar(30),

@addPublisher nvarchar(30),

@phoneNumber varchar(10)

as

begin

insert into dbo.PUBLISHER

(

idPublisher,

namePublisher,

addressPublisher,

phoneNumber

)

values

(

@idPublisher,

@namePublisher,

@addPublisher,

@phoneNumber

)

End

#### Procedure thêm thông tin loại khách hàng mới

create or alter procedure proc\_addNewTypeCustomer

@idTypeCustomer varchar(8) ,

@nameTypeCus nvarchar(30),

@pointMark int,

@valueTypeCus int

as

begin

insert into dbo.TYPECUSTOMER

(

idTypeCus,

nameTypeCus,

pointMark,

valueTypeCus

)

values

(

@idTypeCustomer,

@nameTypeCus,

@pointMark,

@valueTypeCus

)

end

#### Procedure thêm thông tin một loại voucher

create or alter procedure proc\_addNewVoucher

@idVoucher varchar(8),

@valueVoucher int,

@nameOfEvent nvarchar(40),

@dateStart date,

@dateEnd date,

@amount int

as

begin

insert into dbo.VOUCHER

(

idVoucher,

valueVoucher,

nameOfEventVoucher,

dateStart,

dateEnd,

amount

)

values

(

@idVoucher,

@valueVoucher,

@nameOfEvent,

@dateStart,

@dateEnd,

@amount

)

End

#### Procedure thêm một thông tin khách hàng mới

create or alter procedure proc\_addNewCustomer

@idCustomer varchar(8) ,

@nameCus nvarchar(30),

@addressCus nvarchar(30),

@phoneNumber varchar(20),

@idTypeCus varchar(8)

as

begin

insert into dbo.CUSTOMER

(

idCus,

nameCus,

addressCus,

phoneNumber,

idTypeCus

)

values

(

@idCustomer,

@nameCus,

@addressCus,

@phoneNumber,

@idTypeCus

)

End

### 3.2 Procedure chỉnh sửa dữ liệu ở các bảng

#### 3.2.1. Procedure chỉnh sửa thông tin tài khoản

create or alter procedure proc\_updateAccount

@idAccount varchar(8) ,

@nameAccount varchar(20),

@password varchar(30)

as

begin

begin transaction updateAcc

begin try

exec proc\_updateUser @nameAccount, @password

--Chỉ thay đổi mật khẩu

update dbo.ACCOUNT

set ACCOUNT.password=@password

where ACCOUNT.idAccount=@idAccount

commit transaction udpateAcc

end try

begin catch

print (error\_message())

rollback transaction updateAcc

end catch

end

#### 3.2.2. Procedure chỉnh sửa thông tin tác giả

create or alter procedure proc\_updateAuthor

@idAuthor varchar(8) ,

@nameAuthor nvarchar(30),

@phoneNumber varchar(10)

as

begin

update dbo.AUTHOR

set AUTHOR.nameAuthor=@nameAuthor, AUTHOR.phoneNumber=@phoneNumber

where dbo.AUTHOR.idAuthor=@idAuthor

end

#### 3.2.3. Procedure chỉnh sửa thông tin loại sách

create or alter procedure proc\_updateBook

@idBook varchar(8) ,

@nameBook nvarchar(20),

@urlImage image,

@priceImport int,

@priceExport int,

@idPublisher varchar(8)

as

begin

update dbo.BOOK

set BOOK.nameBook=@nameBook, BOOK.urlImage=@urlImage, BOOK.priceImport=@priceImport, BOOK.priceExport=@priceExport, BOOK.idPublisher=@idPublisher

where dbo.BOOK.idBook=@idBook

end

#### 3.2.4. Procedure chỉnh sửa thông tin tác giả cho loại sách

create or alter procedure proc\_updateBookAuthor

@idBook varchar(8) ,

@nameAuthor nvarchar(30),

@nameNewAuthor nvarchar(30)

as

begin

declare @idAuthor varchar(8) , @idNewAuthor varchar(8)

select @idAuthor=dbo.func\_getIDAuthor(@nameAuthor), @idNewAuthor=dbo.func\_getIDAuthor(@nameNewAuthor)

update dbo.BOOK\_AUTHOR

set idAuthor=@idNewAuthor

where dbo.BOOK\_AUTHOR.idAuthor=@idAuthor and dbo.BOOK\_AUTHOR.idBook=@idBook

end

#### 3.2.5. Procedure chỉnh sửa thông tin thể loại cho sách

create or alter procedure proc\_updateBookCategory

@idBook varchar(8) ,

@nameCategory nvarchar(20),

@nameNewCategory nvarchar(20)

as

begin

declare @idCategory varchar(8) , @idNewCategory varchar(8)

select @idCategory=dbo.func\_getIDCategory(@nameCategory), @idNewCategory=dbo.func\_getIDCategory(@nameNewCategory)

update dbo.BOOK\_CATEGORY

set idCategory=@idNewCategory

where dbo.BOOK\_CATEGORY.idCategory=@idCategory and dbo.BOOK\_CATEGORY.idBook=@idBook

end

#### 3.2.6. Procedure chỉnh sửa thông tin thể loại trong dữ liệu

create or alter procedure proc\_updateCategory

@idCategory varchar(8) ,

@nameCategory nvarchar(20),

@describleCategory nvarchar(50)

as

begin

update dbo.CATEGORY

set nameCategory=@nameCategory, describeCategory=@describleCategory

where dbo.CATEGORY.idCategory=@idCategory

end

go

#### 3.2.7. Procedure chỉnh sửa thông tin khách hàng

create or alter procedure proc\_updateCustomer

@idCus varchar(8) ,

@nameCus nvarchar(30),

@addressCus nvarchar(30),

@phoneNumber varchar(20)

as

begin

update dbo.CUSTOMER

set nameCus=@nameCus, addressCus=@addressCus, phoneNumber=@phoneNumber

where dbo.CUSTOMER.idCus=@idCus

end

#### 3.2.8. Procedure chỉnh sửa thông tin nhân viên

create or alter procedure proc\_updateEmployee

@idEmployee varchar(8) ,

@firstName nvarchar(10),

@middleName nvarchar(10),

@lastName nvarchar(10),

@sex nvarchar(5),

@addEmp nvarchar(30),

@phoneNumber varchar(10),

@email varchar(50),

@yearOfBirth date

as

begin

update dbo.EMPLOYEE

set firstName=@firstName, middleName=@middleName, lastName=@lastName, sex=@sex, addEmp=@addEmp, phoneNumber=@phoneNumber, email=@email, yearOfBirth=@yearOfBirth

where dbo.EMPLOYEE.idEmployee=@idEmployee

end

#### 3.2.9. Procedure chỉnh sửa thông tin nhà xuất bản

create or alter procedure proc\_updatePublisher

@idPublisher varchar(8) ,

@namePublisher nvarchar(30),

@addPublisher nvarchar(30),

@phoneNumber varchar(10)

as

begin

update dbo.PUBLISHER

set namePublisher=@namePublisher, addressPublisher=@addPublisher, phoneNumber=@phoneNumber

where dbo.PUBLISHER.idPublisher=@idPublisher

end

#### 3.2.10. Procedure chỉnh sửa thông tin loại khách hàng

create or alter procedure proc\_updateTypeCustomer

@idTypeCus varchar(8) ,

@nameTypeCus nvarchar(30),

@pointMark int,

@valueTypeCus int

as

begin

update dbo.TYPECUSTOMER

set nameTypeCus=@nameTypeCus, pointMark=@pointMark, valueTypeCus=@valueTypeCus

where dbo.TYPECUSTOMER.idTypeCus=@idTypeCus

end

#### 3.2.11 Procedure chỉnh sửa thông tin voucher

create or alter procedure proc\_updateVoucher

@idVoucher varchar(8) ,

@valueVoucher int,

@nameOfEvent nvarchar(40),

@dateStart date,

@dateEnd date,

@amount int

as

begin

update dbo.VOUCHER

set valueVoucher=@valueVoucher, nameOfEventVoucher=@nameOfEvent, dateStart=@dateStart, dateEnd=@dateEnd, amount=@amount

where dbo.VOUCHER.idVoucher=@idVoucher

End

### 3.3 Procedure xóa dữ liệu ở các bảng

#### 3.3.1. Procedure xóa thông tin tài khoản

create or alter procedure proc\_DeleteAccount (@idAccount varchar(8) )

as

begin

declare @username nvarchar (20)

set @username = (select ACCOUNT.nameAccount from ACCOUNT where idAccount = @idAccount)

begin transaction deleteAcc

begin try

-- Xóa user khỏi hệ thống

delete dbo.ACCOUNT

where dbo.ACCOUNT.idAccount=@idAccount

exec proc\_deleteUser @username

commit transaction deleteAcc

end try

begin catch

print error\_message()

rollback transaction deleteAcc

end catch

end

#### 3.3.2. Procedure xóa thông tin tác giả

create or alter procedure proc\_DeleteAuthor

@idAuthor varchar(8)

as

begin

delete from dbo.AUTHOR

where dbo.AUTHOR.idAuthor=@idAuthor

end

#### 3.3.3. Procedure xóa thông tin sách

create or alter procedure proc\_DeleteBook

@idBook varchar(8)

as

begin

delete from dbo.BOOK

where dbo.BOOK.idBook=@idBook

end

#### 3.3.4. Procedure xóa thông tin tác giả của loại sách

create or alter procedure proc\_deleteBookAuthor

@idBook varchar(8) ,

@nameAuthor nvarchar(30)

as

begin

declare @idAuthor varchar(8)

select @idAuthor=dbo.func\_getIDAuthor(@nameAuthor)

delete from dbo.BOOK\_AUTHOR

where dbo.BOOK\_AUTHOR.idBook=@idBook and dbo.BOOK\_AUTHOR.idAuthor=@idAuthor

end

#### 3.3.5. Procedure xóa thông tin thể loại của sách

create or alter procedure proc\_deleteBookCategory

@idBook varchar(8) ,

@nameCategory nvarchar(20)

as

begin

declare @idCategory varchar(8)

select @idCategory=dbo.func\_getIDCategory(@nameCategory)

delete from dbo.BOOK\_CATEGORY

where dbo.BOOK\_CATEGORY.idBook=@idBook and dbo.BOOK\_CATEGORY.idCategory=@idCategory

end

#### 3.3.6. Procedure xóa thông tin thể loại trong dữ liệu

create or alter procedure proc\_deleteCategory

@idCategory varchar(8)

as

begin

delete from dbo.CATEGORY

where dbo.CATEGORY.idCategory=@idCategory

end

#### 3.3.7. Procedure xóa thông tin khách hàng trong dữ liệu

create or alter procedure proc\_deleteCustomer

@idCus varchar(8)

as

begin

delete from dbo.CUSTOMER

where dbo.CUSTOMER.idCus=@idCus

end

#### 3.3.8. Procedure xóa thông tin nhân viên trong dữ liệu

create or alter procedure proc\_deleteEmployee

@idEmployee varchar(8)

as

begin

delete from dbo.EMPLOYEE

where dbo.EMPLOYEE.idEmployee=@idEmployee

end

#### 3.3.9. Procedure xóa thông tin nhà xuất bản trong dữ liệu

create or alter procedure proc\_deletePublisher

@idPublisher varchar(8)

as

begin

delete from dbo.PUBLISHER

where dbo.PUBLISHER.idPublisher=@idPublisher

end

#### 3.3.10. Procedure xóa thông tin loại khách hàng trong dữ liệu

create or alter procedure proc\_deleteTypeCustomer

@idTypeCus varchar(8)

as

begin

delete from dbo.TYPECUSTOMER

where dbo.TYPECUSTOMER.idTypeCus=@idTypeCus

end

#### 3.3.11. Procedure xóa thông tin loại voucher trong dữ liệu

create or alter procedure proc\_deleteVoucher

@idVoucher varchar(8) as

begin

delete from dbo.VOUCHER

where dbo.VOUCHER.idVoucher=@idVoucher

end

### 3.4 Các chức năng thao tác nhập với sách

#### 3.4.1. Procedure tạo hóa đơn nhập sách

create or alter procedure proc\_addNewBillInput

@idBillInput varchar(8)

as

begin

insert into dbo.BILLINPUT

(

idBillInput

)

values

(

@idBillInput

)

End

#### 3.4.2. Procedure thêm sách vào đơn hàng

create or alter procedure proc\_addNewBookBillInput

@idBillInput varchar(8),

@idBook varchar(8),

@amountInput int

as

begin

begin transaction

begin try

insert into dbo.BOOK\_BILLINPUT

(

idBillInput,

idBook,

amountInput

)

values

(

@idBillInput,

@idBook,

@amountInput

)

--Cập nhật lại số lượng sách trong cửa hàng

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount+@amountInput

where dbo.BOOK.idBook=@idBook

--Cập nhật tổng tiền hóa đơn

declare @totalBill int

select @totalBill=dbo.func\_returnToTalOfBillInput(@idBillInput)

update dbo.BILLINPUT

set dbo.BILLINPUT.total=@totalBill

where dbo.BILLINPUT.idBillInput=@idBillInput

commit transaction

end try

begin catch

raiserror('Sách đã có trong đơn hàng. Vui lòng thay đổi số lượng trong đơn',16,1)

rollback transaction

end catch

end

3.4.3. Procedure sửa số lượng sách trong đơn hàng

create or alter procedure proc\_updateBookBillInput

@idBillImport varchar(8),

@idBook varchar(8),

@amountBook int,

@amountNewBook int

as

begin

begin transaction

begin try

--Cập nhật lại thông tin trong giỏ hàng

update dbo.BOOK\_BILLINPUT

set amountInput=@amountNewBook

where dbo.BOOK\_BILLINPUT.idBook=@idBook and dbo.BOOK\_BILLINPUT.idBillInput=@idBillImport

--Cập nhật lại số lượng sách trong kho

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount-@amountBook+@amountNewBook

where dbo.BOOK.idBook=@idBook

commit transaction

end try

begin catch

raiserror('Đã có lỗi xảy ra. Vui lòng thử lại',16,1)

rollback transaction

end catch

end

#### 3.4.4. Procedure xóa một item trong đơn hàng nhập

create or alter procedure proc\_deleteBookBillInput

@idBillInput varchar(8) ,

@idBook varchar(8)

as

begin

--Lấy ra số lượng sách có trong hóa đơn

declare @amountBook int

select @amountBook=dbo.BOOK\_BILLINPUT.amountInput

from dbo.BOOK\_BILLINPUT

where dbo.BOOK\_BILLINPUT.idBook=@idBook and dbo.BOOK\_BILLINPUT.idBillInput=@idBillInput

--Xóa sách đó trong hóa đơn

delete from dbo.BOOK\_BILLINPUT

where dbo.BOOK\_BILLINPUT.idBillInput=@idBillInput and dbo.BOOK\_BILLINPUT.idBook=@idBook

--Cập nhật lại số lượng trong kho

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount-@amountBook

where dbo.BOOK.idBook=@idBook

end

#### 3.4.5. Procedure xác nhận xuất hóa đơn

create or alter procedure proc\_confirmBillImport

@idBillInput varchar(8) ,

@dateTimeOfBill date,

@idEmp varchar(8)

as

begin

declare @totalOfBill int

select @totalOfBill=dbo.BILLINPUT.total

from dbo.BILLINPUT

where dbo.BILLINPUT.idBillInput=@idBillInput

if(@totalOfBill<=0)

begin

delete from dbo.BILLINPUT

where dbo.BILLINPUT.idBillInput=@idBillInput

return;

end

update dbo.BILLINPUT

set dateOfInput=@dateTimeOfBill, idEmployee=@idEmp

where dbo.BILLINPUT.idBillInput=@idBillInput

end

#### 3.4.6. Procedure xác nhận hủy hóa đơn hiện tại

create or alter procedure proc\_cancelBillImport

@idBill varchar(8)

as

begin

--Duyệt qua từng idBook trong hóa đơn

declare @idBook varchar(8)

declare item cursor for (select dbo.BOOK\_BILLINPUT.idBook from dbo.BOOK\_BILLINPUT where dbo.BOOK\_BILLINPUT.idBillInput=@idBill)

open item

fetch next from item into @idBook

while @@FETCH\_STATUS=0

begin

execute proc\_deleteBookBillInput @idBill, @idBook

fetch next from item into @idBook

end

close item

deallocate item

--Xóa hóa đơn đó

delete from dbo.BILLINPUT

where dbo.BILLINPUT.idBillInput=@idBill

end

### 3.5 Các chức năng thao tác bán sách

#### 3.5.1. Procedure thêm một hóa đơn bán sách

create or alter procedure proc\_addNewBillOutput

@idBillOutput varchar(8)

as

begin

insert into dbo.BILLOUTPUT

(

idBillOutPut

)

values

(

@idBillOutput

)

End

#### 3.5.2. Procedure thêm một sản phẩm vào hóa đơn bán sách

create or alter procedure proc\_addNewBookBillOutput

@idBillOutput varchar(8) ,

@idBook varchar(8) ,

@amount int

as

begin

--Thêm item vào bảng (amount đảm bảo thỏa ràng buộc do đã có trigger)

if(exists(select \* from dbo.BOOK\_BILLOUTPUT

where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput and dbo.BOOK\_BILLOUTPUT.idBook=@idBook))

begin

update dbo.BOOK\_BILLOUTPUT

set amountOutput=amountOutput+@amount

where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput and dbo.BOOK\_BILLOUTPUT.idBook=@idBook

end

else

begin

insert into dbo.BOOK\_BILLOUTPUT

(

idBillOutput,

idBook,

amountOutput

)

values

(

@idBillOutput,

@idBook,

@amount

)

end

--Cập nhật lại số lượng sách trong giỏ hảng

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount-@amount

where dbo.BOOK.idBook=@idBook

--Cập nhật tổng tiền hóa đơn

declare @totalBill int

select @totalBill=dbo.func\_returnToTalOfBillOutput(@idBillOutput)

update dbo.BILLOUTPUT

set dbo.BILLOUTPUT.total=@totalBill

where dbo.BILLOUTPUT.idBillOutPut=@idBillOutput

end

#### 3.5.3. Procedure cập nhật số lượng mới cho sản phẩm trong đơn

create or alter procedure proc\_updateBookBillOutput

@idBillOutput varchar(8),

@idBook varchar(8),

@amountBook int,

@amountNewBook int

as

begin

begin transaction

begin try

--Cập nhật lại thông tin trong giỏ hàng

update dbo.BOOK\_BILLOUTPUT

set amountOutput=@amountNewBook

where dbo.BOOK\_BILLOUTPUT.idBook=@idBook and dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput

--Cập nhật lại số lượng sách trong kho

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount+@amountBook-@amountNewBook

where dbo.BOOK.idBook=@idBook

commit transaction

end try

begin catch

raiserror('Đã có lỗi xảy ra. Vui lòng thử lại',16,1)

rollback transaction

end catch

end

#### 3.5.4. Procedure xóa một sản phẩm trong giỏ hàng

create or alter procedure proc\_deleteBookBillOutput

@idBillOutput varchar(8) ,

@idBook varchar(8)

as

begin

--Lấy ra số lượng sách có trong hóa đơn

declare @amountBook int

select @amountBook=dbo.BOOK\_BILLOUTPUT.amountOutput

from dbo.BOOK\_BILLOUTPUT

where dbo.BOOK\_BILLOUTPUT.idBook=@idBook and dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput

--Xóa sách đó trong hóa đơn

delete from dbo.BOOK\_BILLOUTPUT

where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput and dbo.BOOK\_BILLOUTPUT.idBook=@idBook

--Cập nhật lại số lượng trong kho

update dbo.BOOK

set dbo.BOOK.amount=dbo.BOOK.amount+@amountBook

where dbo.BOOK.idBook=@idBook

end

#### 3.5.5. Procedure cập nhật loại điểm tích lũyvà up VIP cho khách hàng sau mỗi lần mua

create or alter procedure proc\_updateTypeCusForCus

@amountBooksBought int,

@idCus varchar(8)

as

begin

declare @idTypeCus varchar(8) , @pointMark int, @maxPointValue int

set @maxPointValue=0

declare point cursor for select dbo.TYPECUSTOMER.pointMark from dbo.TYPECUSTOMER order by pointMark ASC

open point

fetch next from point into @pointMark

while @@FETCH\_STATUS=0

begin

if(@amountBooksBought>=@pointMark)

set @maxPointValue=@pointMark

else

break

fetch next from point into @pointMark

end

close point

deallocate point

select @idTypeCus=(select dbo.TYPECUSTOMER.idTypeCus from dbo.TYPECUSTOMER where dbo.TYPECUSTOMER.pointMark=@maxPointValue)

--Update lại IDtypecus cho khách hàng

update dbo.CUSTOMER

set idTypeCus=@idTypeCus, pointCus=@amountBooksBought

where dbo.CUSTOMER.idCus=@idCus

end

#### 3.5.6. Procedure xác nhận xuất hóa đơn

create or alter procedure proc\_confirmBillExport

@idBillOutput varchar(8),

@dateTimeOfBill date,

@idCus varchar(8) ,

@idEmp varchar(8) ,

@idVoucher varchar(8) = null

as

begin

declare @totalOfBill int , @discountOfVoucher int, @discountOfTypeCus int

--Lấy ra giá trị hóa đơn và giá trị chiết khấu

select @totalOfBill=dbo.BILLOUTPUT.total

from dbo.BILLOUTPUT

where dbo.BILLOUTPUT.idBillOutPut=@idBillOutput

--Lấy giá trị chiết khấu của loại khách hàng

set @discountOfTypeCus=dbo.func\_getValueDiscountOfTypeCustomer(@idCus)

--Tính giá trị chiết khấu voucher nếu có

if(@idVoucher is not null)

begin

select @discountOfVoucher=dbo.VOUCHER.valueVoucher

from dbo.VOUCHER

where dbo.VOUCHER.idVoucher=@idVoucher

end

else

set @discountOfVoucher=0

--Tính tiền giá trị hóa đơn hiện tại

set @totalOfBill=@totalOfBill-((@discountOfTypeCus\*@totalOfBill)/100)

set @totalOfBill=@totalOfBill-@discountOfVoucher

--Nếu tiền hóa đơn bé hơn hoặc bằng 0 thì hủy bỏ hóa đơn đó

if(@totalOfBill<=0)

begin

raiserror('Hóa đơn không hợp lệ',16,1)

delete from dbo.BILLOUTPUT

where dbo.BILLOUTPUT.idBillOutPut=@idBillOutput

return;

end

--Cập nhật hóa đơn vào bảng

update dbo.BILLOUTPUT

set dateOfBill=@dateTimeOfBill, idCus=@idCus, idEmployee=@idEmp, idVoucher=@idVoucher, total=@totalOfBill

where dbo.BILLOUTPUT.idBillOutPut=@idBillOutput

--Cập lại điểm tích lũy và Type cho khách hàng

declare @amountBookInBill int, @amountBookTotal int

select @amountBookInBill=(select sum(amountOutput) from dbo.BOOK\_BILLOUTPUT where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBillOutput)

select @amountBookTotal=@amountBookInBill+(select dbo.CUSTOMER.pointCus from dbo.CUSTOMER where dbo.CUSTOMER.idCus=@idCus)

exec proc\_updateTypeCusForCus @amountBooksBought=@amountBookTotal, @idCus=@idCus

end

#### 3.5.7. Procedure xác nhận hủy giỏ hàng hiện tại

create or alter procedure proc\_cancelBillExport

@idBill varchar(8)

as

begin

--Duyệt qua từng idBook trong hóa đơn

declare @idBook varchar(8)

declare item cursor for (select dbo.BOOK\_BILLOUTPUT.idBook from dbo.BOOK\_BILLOUTPUT where dbo.BOOK\_BILLOUTPUT.idBillOutput=@idBill)

open item

fetch next from item into @idBook

while @@FETCH\_STATUS=0

begin

execute proc\_deleteBookBillOutput @idBill, @idBook

fetch next from item into @idBook

end

close item

deallocate item

--Xóa hóa đơn đó

delete from dbo.BILLOUTPUT

where dbo.BILLOUTPUT.idBillOutPut=@idBill

end

### 3.6 Các thống kê doanh thu của cửa hàng

#### 3.6.1. Procedure show doanh thu trong 2 mốc thời gian do người dùng nhập

create or alter proc sp\_ShowRevenue

@begin date , @end date

as

begin

select convert(date,dateOfBill) as dateOfBill, sum(total) as total

from BILLOUTPUT where dateOfBill between @begin and @end

group by convert(date,dateOfBill)

end

3.6.2. Procedure show top 5 sản phẩm bán chạy nhất trong 2 mốc thời gian do người dùng nhập

create or alter proc sp\_ShowTop5Book

@begin date , @end date

as

begin

select top 5 BOOK.nameBook, sum(BOOK\_BILLOUTPUT.amountOutput) as amountOutput

from BOOK\_BILLOUTPUT, BOOK, BILLOUTPUT

where BOOK\_BILLOUTPUT.idBook= BOOK.idBook

and BILLOUTPUT.idBillOutPut= BOOK\_BILLOUTPUT.idBillOutput

and dateOfBill between @begin and @end

group by BOOK.nameBook

order by amountOutput desc

end

#### 3.6.3. Procedure tổng quan về tổng doanh thu của cửa hàng

create or alter proc sp\_Overview\_Revenue

@begin date, @end date

as

begin

select sum(total)

from BILLOUTPUT

where dateOfBill between @begin and @end

end

#### 3.6.4. Procedure tổng quan về số lượng các đơn hàng

create or alter proc sp\_Overview\_AmountBillOutput

@begin date, @end date

as

begin

select count(\*)

from BILLOUTPUT

where dateOfBill between @begin and @end

end

#### 3.6.5. Procedure tổng quan về số lượng sách bán được

create or alter proc sp\_Overview\_AmountBookBillOutput

@begin date, @end date

as

begin

select sum(amountOutput)

from BOOK\_BILLOUTPUT, BILLOUTPUT,BOOK

where BOOK\_BILLOUTPUT.idBook= BOOK.idBook

and BILLOUTPUT.idBillOutPut= BOOK\_BILLOUTPUT.idBillOutput

and dateOfBill between @begin and @end

end

## Danh sách các View của đề tài

### 4.1. View hiển thị ra thông tin tài khoản

View hiển thị thông tin tài khoản gồm Mã tài khoản, tên tài khoản, mật khẩu và loại tài khoản admin hoặc nhân viên

create or alter view view\_Account

as

select idaccount,nameaccount,password,case when dbo.ACCOUNT.typeOfAcc=1

then

'Admin'

else

'Nhân viên'

end as TypeAcc,idemployee

### 4.2. View hiển thị thông tin tác giả

create or alter view view\_Author

as

select \*

from dbo.AUTHOR

### 4.3. View hiển thị thông tin nhà xuất bản

create or alter view view\_Publisher

as

select \*

from dbo.PUBLISHER

### 4.5. View hiển thị thông tin sách

create or alter view view\_Book

as

select \*

from dbo.BOOK

### 4.6. View hiển thị thông tin thể loại

create or alter view view\_Category

as

select \*

from dbo.CATEGORY

### 4.7. View hiển thị thông tin hóa đơn nhập

create or alter view view\_ReceiptImport

as

select \*

from dbo.BILLINPUT

### 4.8. View hiển thị thông tin hóa đơn xuất

create or alter view view\_ReceiptExport

as

select \*

from dbo.BILLOUTPUT

### 4.9. View hiển thị thông tin khách hàng

create or alter view view\_Customer

as

select \*

from dbo.CUSTOMER

### 4.10. View hiển thị tất cả thông tin loại khách hàng

create or alter view view\_TypeCustomer

as

select \*

from dbo.TYPECUSTOMER

### 4.11. View hiển thị toàn bộ thông tin nhân viên

create or alter view view\_Employee

as

select \*

from dbo.EMPLOYEE

### 4.12. View hiển thị toàn bộ thông tin voucher

create or alter view view\_Voucher

as

select \*

from dbo.VOUCHER

### 4.13. View hiển thị danh sách đang bán trong cửa hàng

View hiển thị danh sách sản phẩm đang bán trong cửa hàng gồm mã ID, tên sách, số lượng và giá bán

create or alter view view\_GetAllBookForSale

as

select BOOK.idBook as ID, BOOK.nameBook as N'Tên Sách', BOOK.amount as N'Số lượng', BOOK.priceExport as N'Giá'

from dbo.BOOK

### 4.14 View hiển thị doanh thu top 5 sản phẩm bán chạy nhất

View hiển thị danh sách doanh thu của top 5 sản phẩm bán chạy nhất gồm tên sản phẩm và số lượng bán

create view view\_top5StockMin

as select top 5 nameBook,amount from BOOK order by amount asc

## Tạo user và phân quyền viết dưới dạng procedure

### 5.1. Procedure tạo user

create or alter proc proc\_createUser (@user nvarchar(30), @pass nvarchar (30))

as

begin

declare @createLogin nvarchar (100)

declare @createUser nvarchar (100)

set @createLogin = 'Create Login ' + @user + ' with password = ''' + @pass + ''''

set @createUser = 'Create User ' + @user + ' For Login ' + @user

print @createLogin

print @createUser

exec (@createLogin)

exec (@createUser)

exec proc\_permission @user, @pass

end

Go

### 5.2. Procedure cập nhật tài khoản user

Create or alter proc proc\_updateUser (@user varchar (20), @newPass varchar(30), @oldPass varchar)

as

begin

declare @state varchar (200)

set @state = 'login '+@user+' with password = '+@newPass --+ ' old\_password = ' +@oldPass

exec (@state)

end

### 5.3. procedure xóa user

create or alter proc proc\_deleteUser (@user nvarchar(20))

as

begin

declare @state1 nvarchar(200), @state2 nvarchar(200)

begin transaction deleteUser

begin try

set @state1 = 'drop login ' +@user

set @state2 = 'drop user ' +@user

exec (@state1)

exec (@state2)

commit transaction deleteUser

end try

begin catch

print (error\_message())

### 5.4. procedure phân quyền cho admin và nhân viên

create or alter proc proc\_permission (@user nvarchar (20), @pass nvarchar(20))

as

begin

declare @state nvarchar (100), @role bit

set @role = (select typeOfAcc from ACCOUNT where nameAccount like @user and password like @pass)

print @role

if (@role = 1)

begin

set @state = 'grant exec, control to ' + @user

exec (@state)

exec master ..sp\_addsrvrolemember @user, N'sysadmin'

print N'Đã gán quyền của admin'

end

else if(@role = 0)

begin

--cấp quyền vào các bảng

set @state = 'grant select, update, delete, insert to '+ @user

exec (@state)

set @state = 'grant exec to '+ @user

exec (@state)

-- cấm quyền vào account

set @state = 'deny select, update, insert, delete on object::ACCOUNT To ' + @user

exec (@state)

-- cấm quyền xem account

set @state = 'deny select, update, insert, delete on object::view\_Account to ' + @user

exec (@state)

-- cấm quyền thêm account

set @state = 'deny exec on object::proc\_addNewAccount to ' + @user

exec (@state)

-- cấm quyền chỉnh sửa account

set @state = 'deny exec on object::proc\_updateAccount to ' + @user

exec (@state)

-- cấm quyền xóa account

set @state = 'deny exec on object::proc\_DeleteAccount to ' + @user

exec (@state)

-- cấm quyền tìm kiếm account

set @state = 'deny select, update, insert, delete on object::func\_searchAccount to ' + @user

exec (@state)

-- cấm quyền vào nhân viên

set @state = 'deny select, update, insert, delete on object::EMPLOYEE To ' + @user

exec (@state)

--cấm quyền xem nhân viên

set @state = 'deny select, update, insert, delete on object::view\_Employee to ' + @user

exec (@state)

-- Cấm quyền thêm nhân viên

set @state = 'deny exec on object::proc\_addNewEmployee to ' + @user

exec (@state)

-- Cấm quyền sửa nhân viên

set @state = 'deny exec on object::proc\_updateEmployee to ' + @user

exec (@state)

-- Cấm quyền xóa nhân viên

set @state = 'deny exec on object::proc\_deleteEmployee to ' + @user

exec (@state)

print N'Đã gán quyền user'

end

End

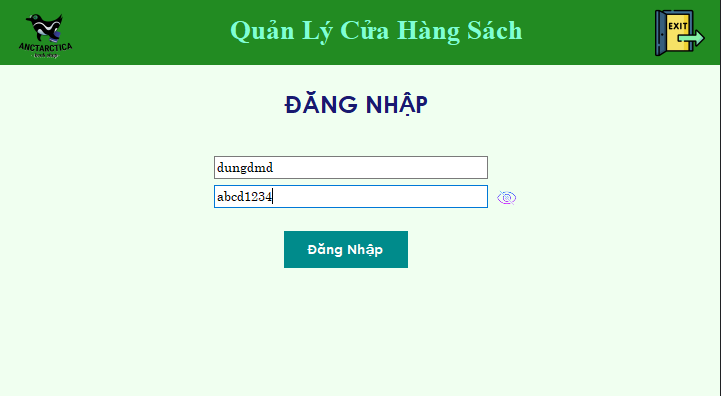
## CHƯƠNG 4: THIẾT KẾ GIAO DIỆN WINFORM

## Giao diện form đăng nhập

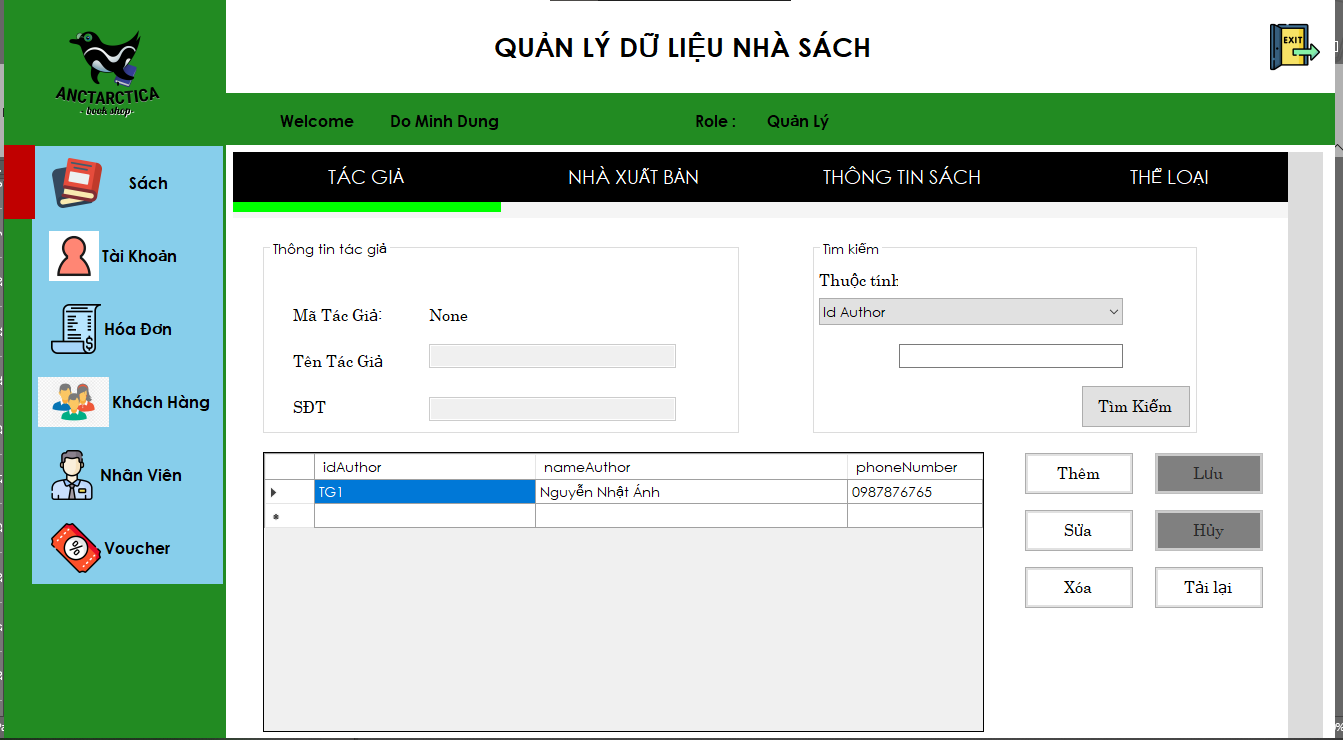
### Đăng nhập với quyền admin

Khi đăng nhập với quyền của admin, người dùng có thể sử dụng đầy đủ các chức năng như xem, thêm, sửa, xóa dữ liệu ở các quan hệ.

Có thể tạo một tài khoản khác, thay đổi mật khẩu hoặc xóa bỏ tài khoản đó.



*Hình 1.1 Giao diện form đăng nhập với admin*



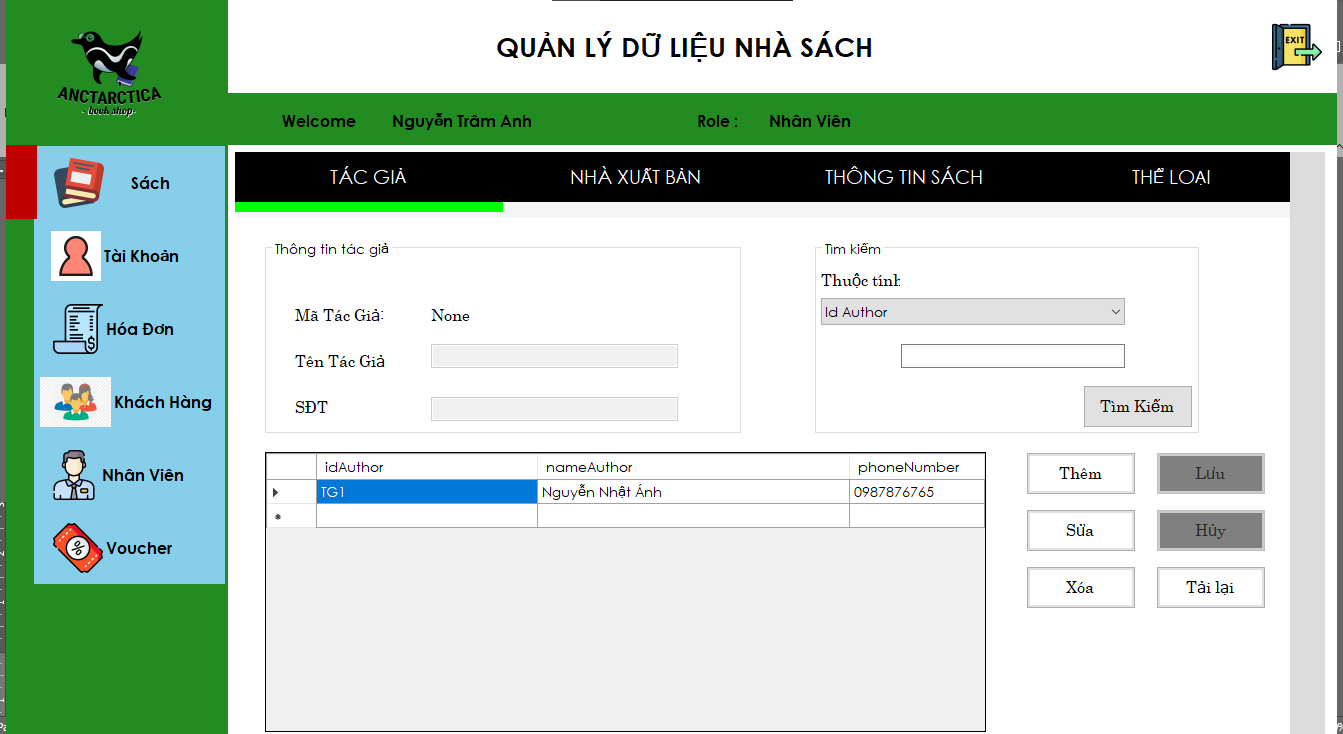
*Hình 1.2 Giao diện sau khi đăng nhập bằng tài khoản admin*

### Đăng nhập với quyền nhân viên

Khi đăng nhập với tài khoản nhận viên. Người sử dụng sẽ không thể thao tác trên các bảng ACCOUNT, EMPLOYEE và không thể vào chức năng thống kê doanh thu.

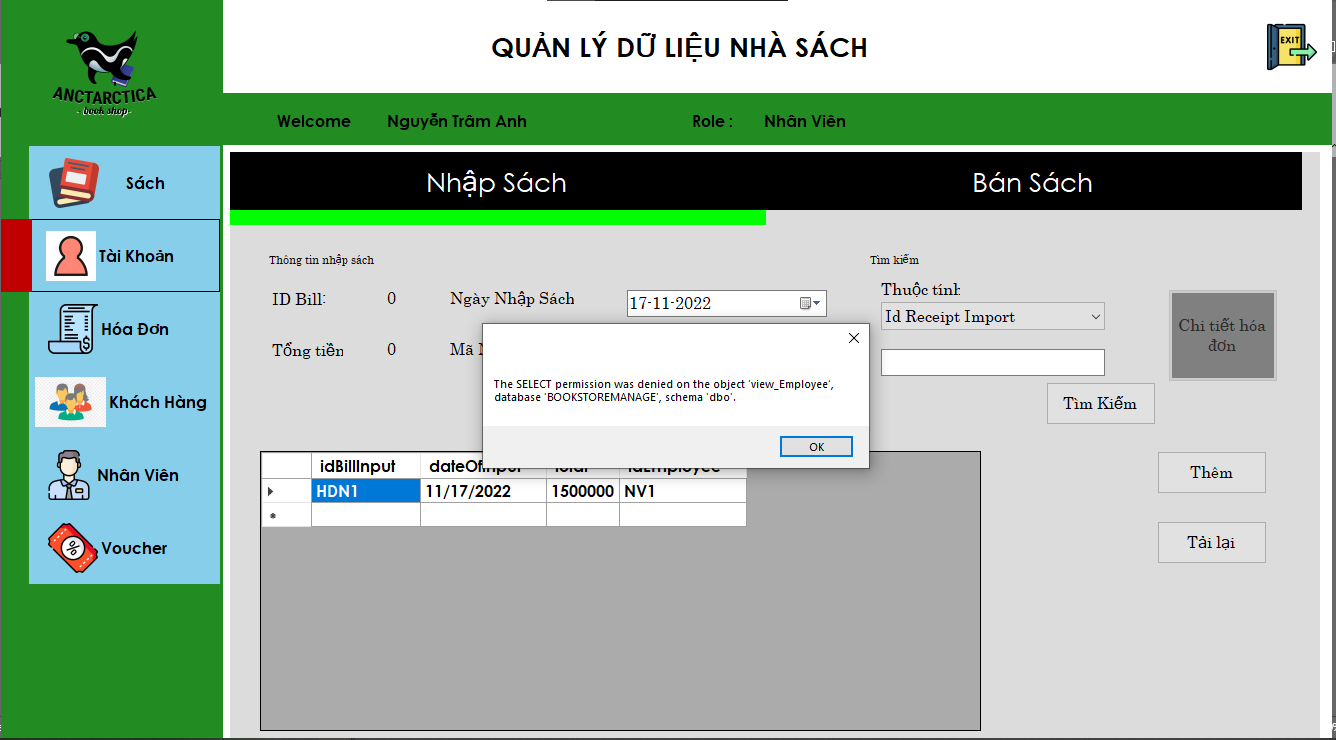


*Hình 1.3 Giao diện Form đăng nhập với nhân viên*

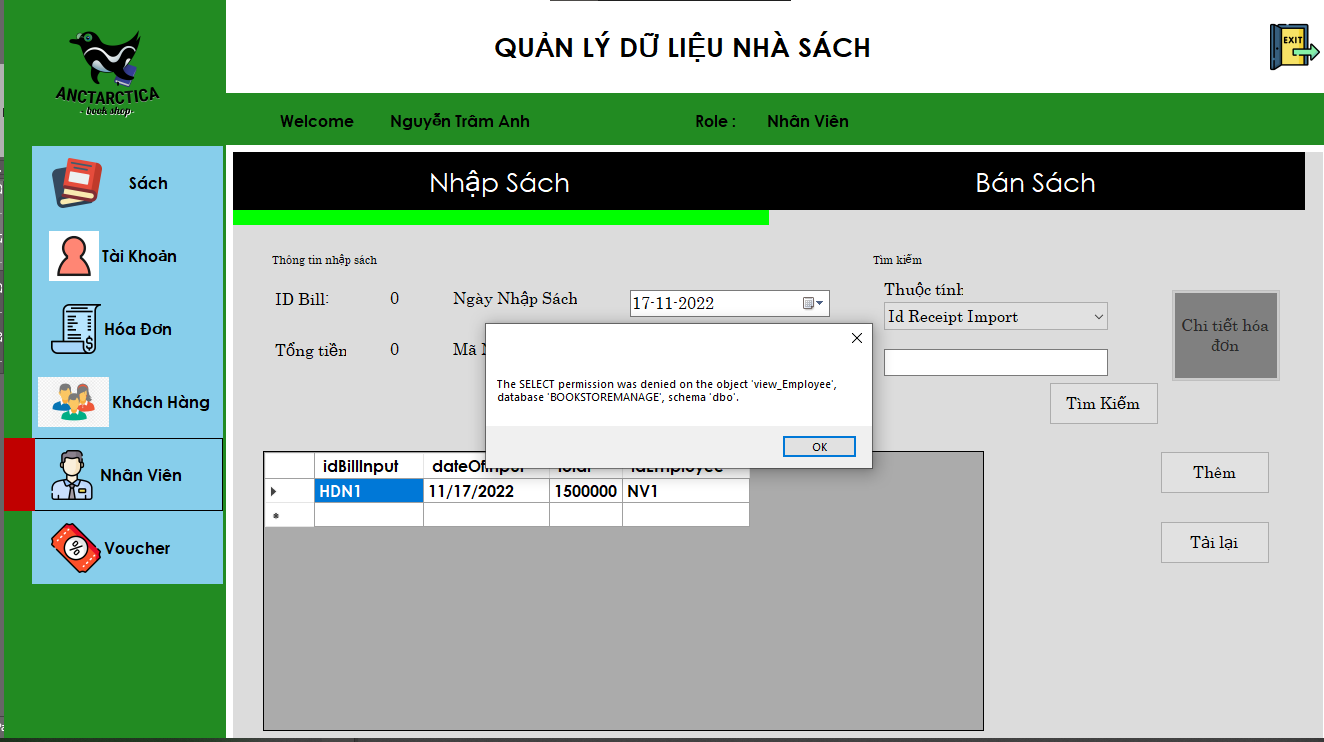


*Hình 1.4 Giao diện sau khi đăng nhập bằng tài khoản nhân viên*

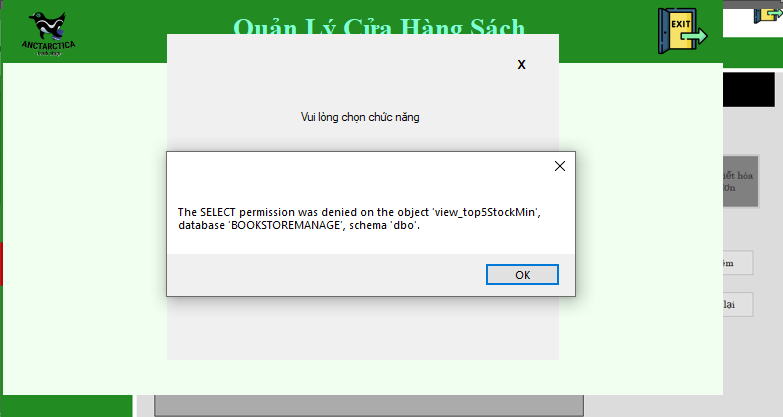
### Các quyền cấm nhân viên không được truy cập vào

**

*Hình 1.5 Nhân viên không thể thao tác trên bảng ACCOUNT*

**

*Hình 1.6 Nhân viên không thể thao tác trên bảng EMPLOYEE*

**

*Hình 1.7 Nhân viên không thể sử dụng chức năng thống kê doanh thu*

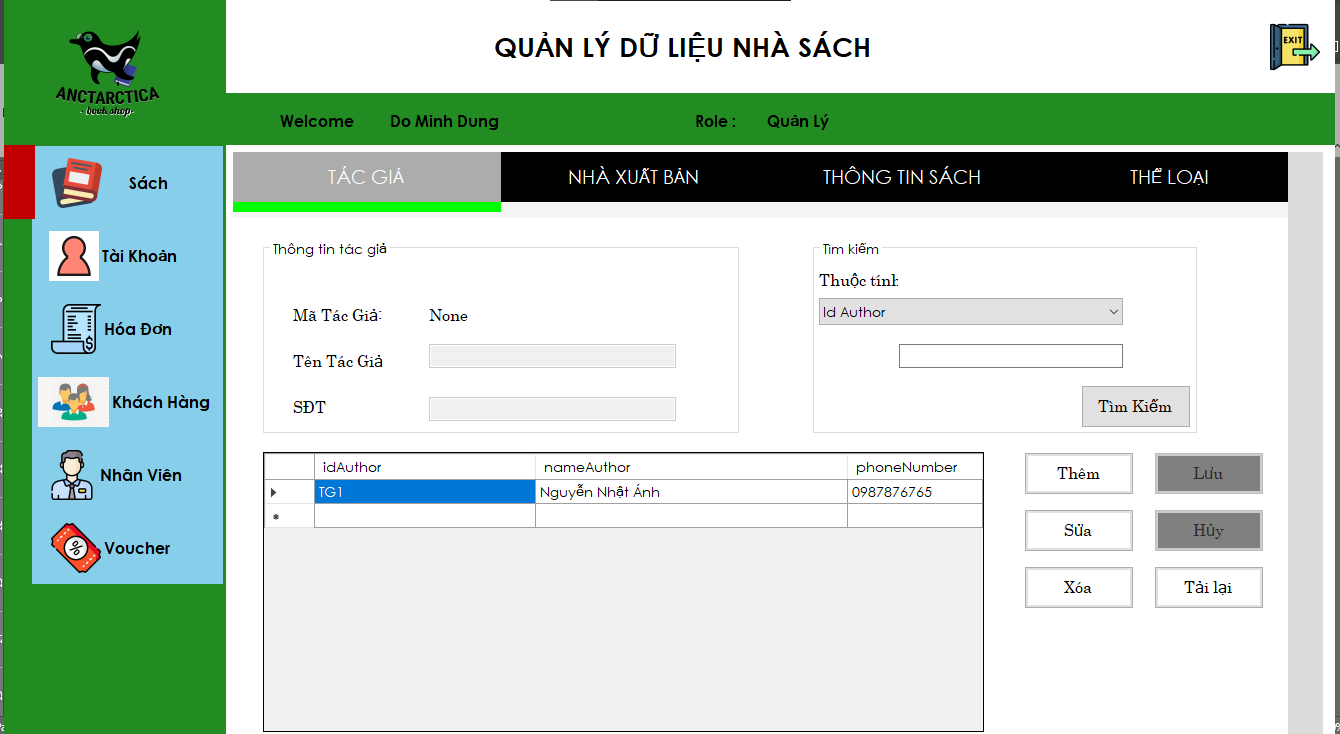
## Giao diện bảng AUTHOR



*Hình 1.8 Giao diện form tác giả*

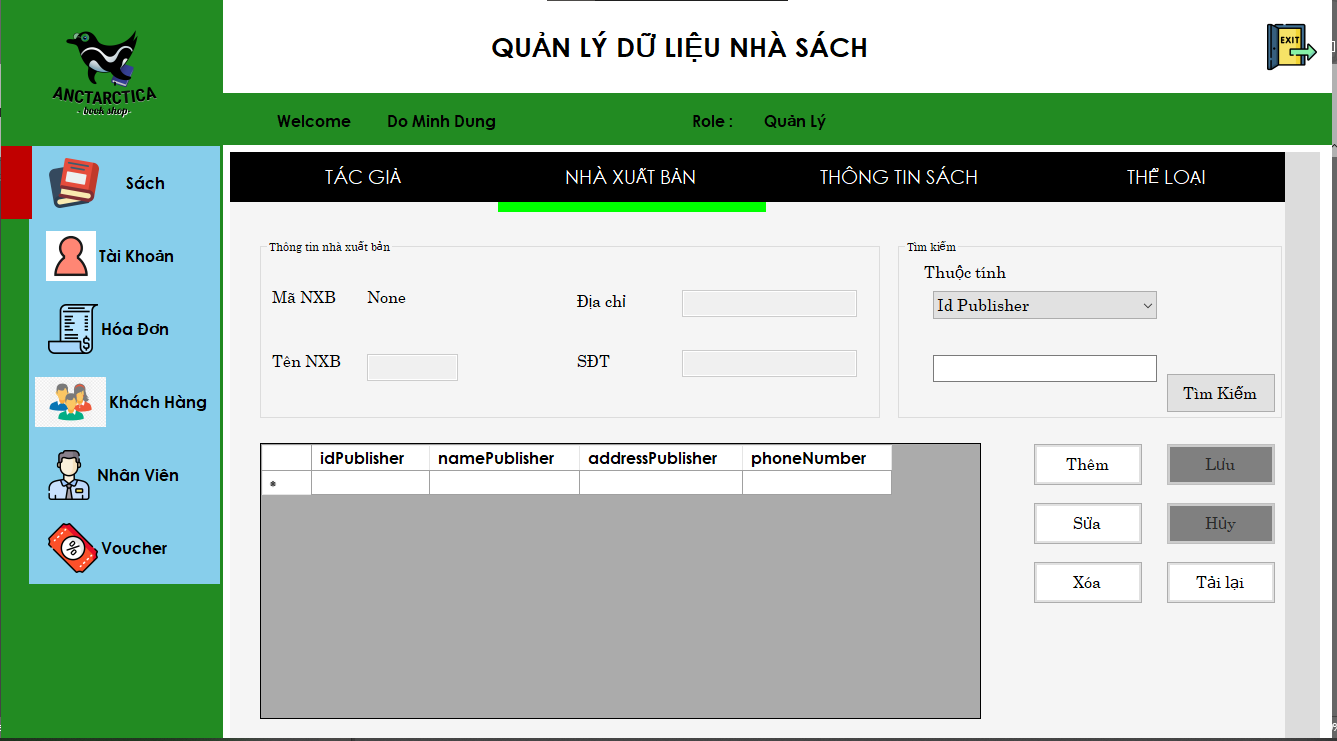
### 2.1 Thêm tác giả

- Giao diện form tác giả sau khi thêm một tác giả mới



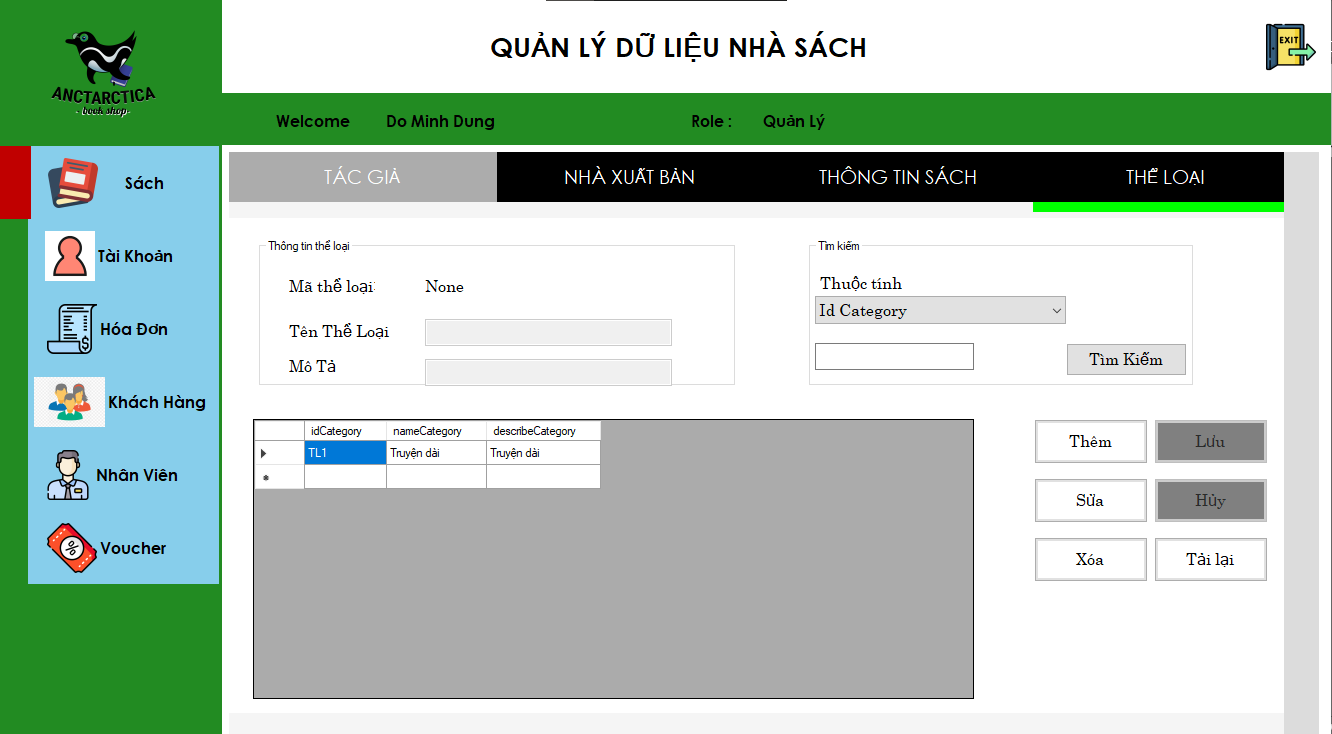
*Hình 1.9 Giao diện sau khi thêm 1 tác giả mới*

## Giao diện form Bảng PUBLISHER

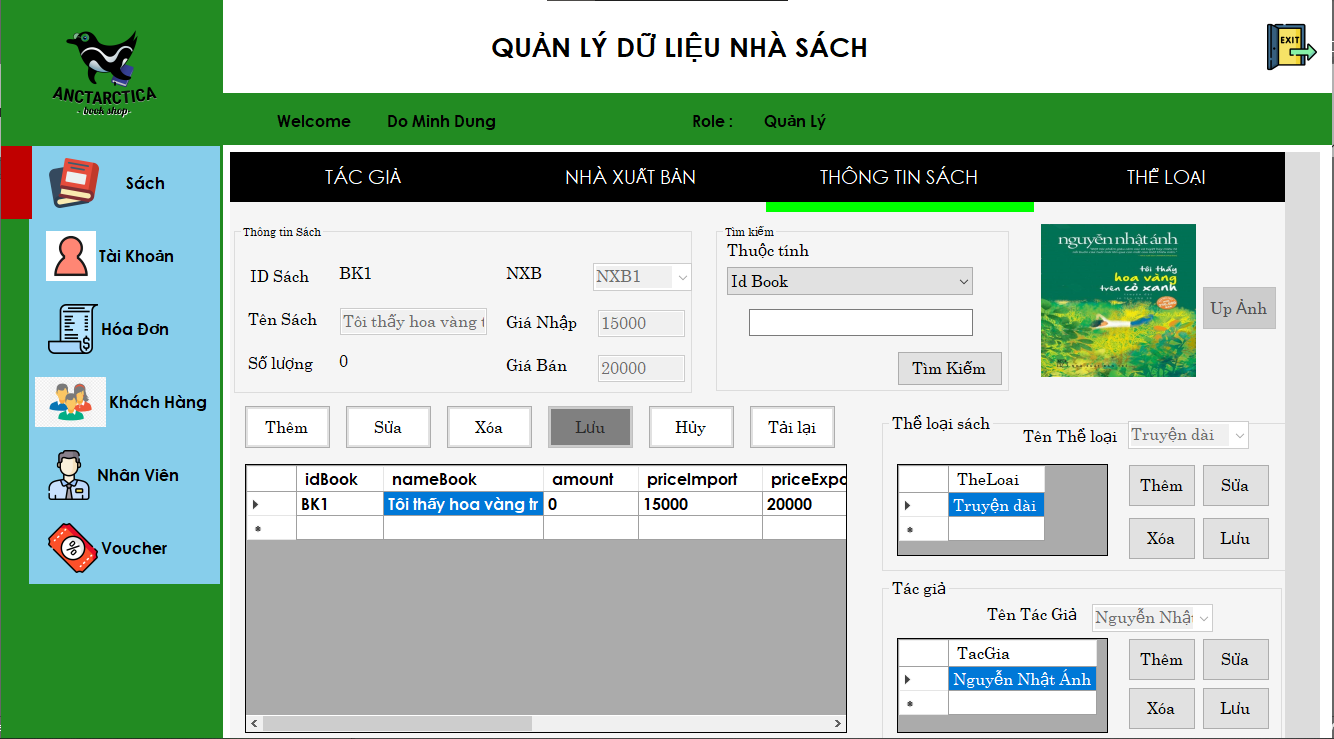


*Hình 1.10 Giao diện form nhà xuất bản*

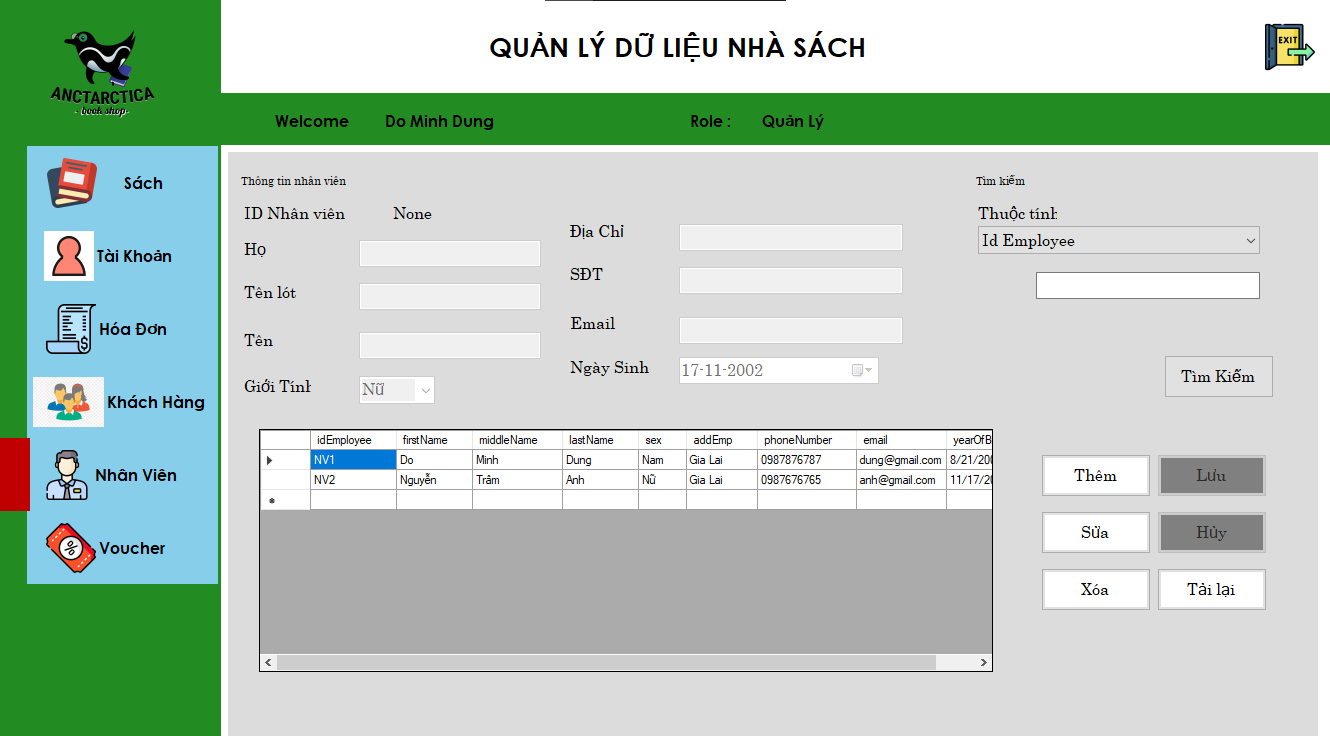
## Giao diện bảng CATEGORY



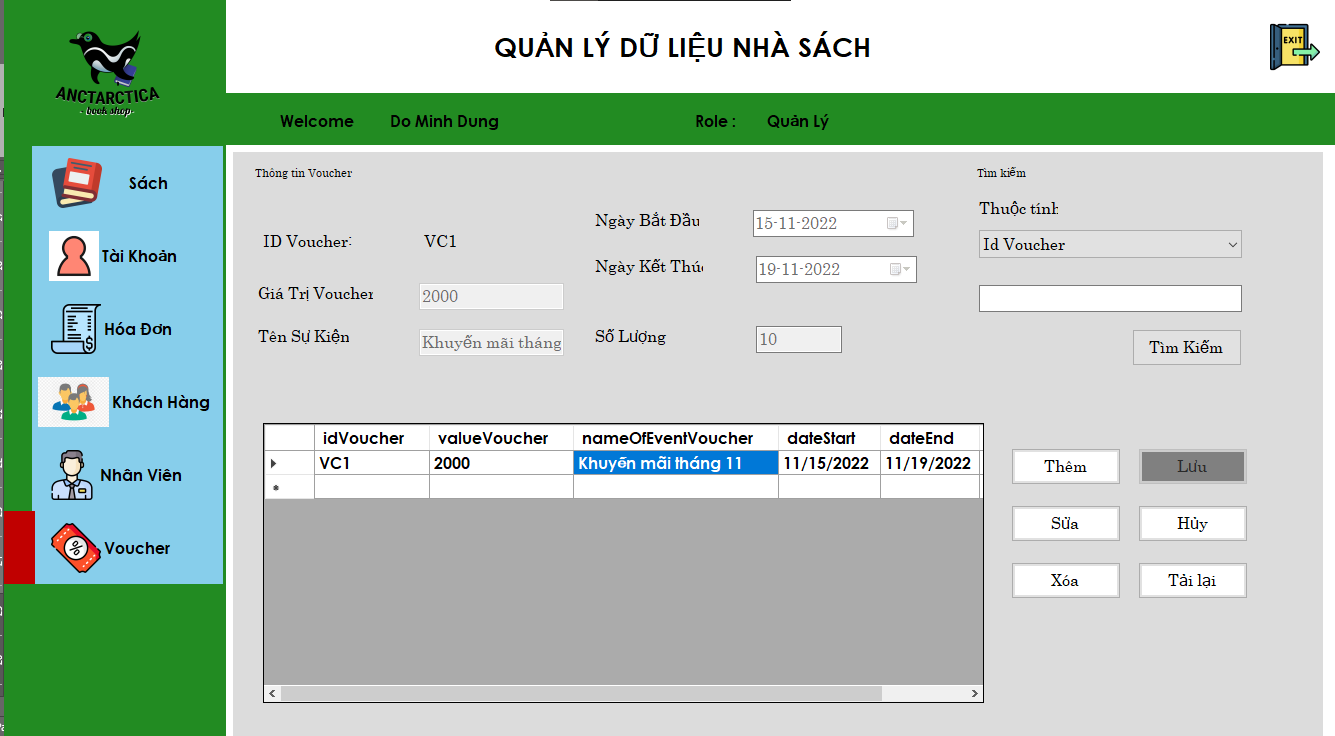
## Giao diện Bảng BOOK



## Giao diện bảng EMPLOYEE



## Giao diện bảng VOUCHER



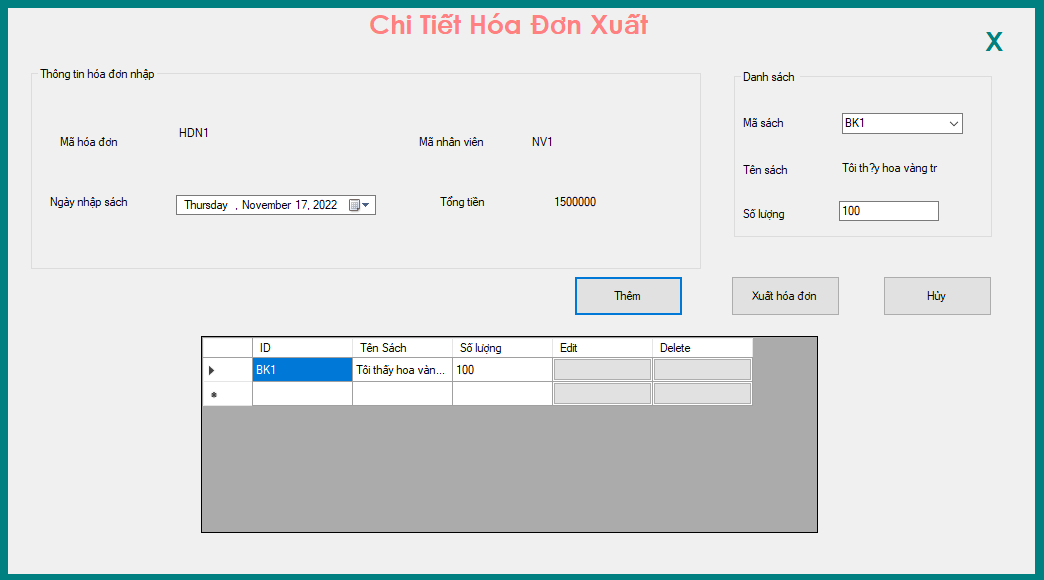
## Giao diện bảng TYPECUSTOMER

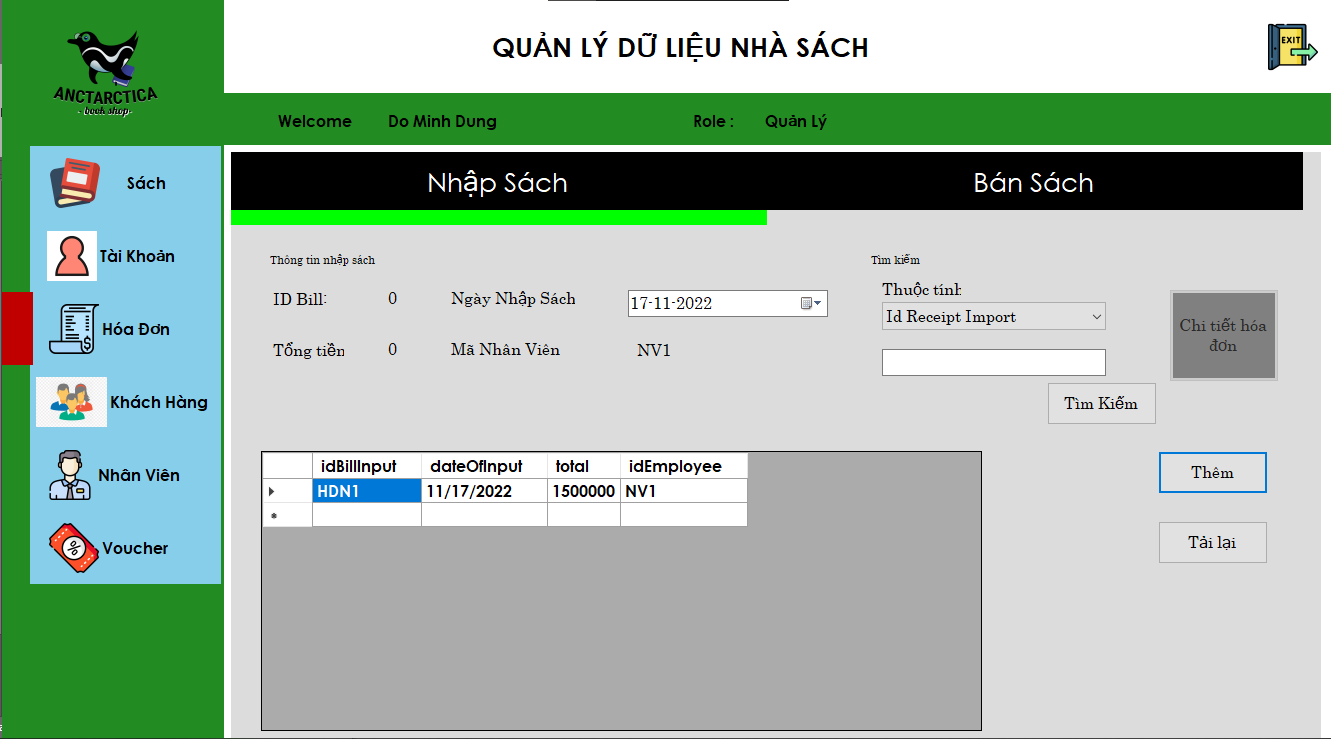


## Giao diện bảng CUSTOMER

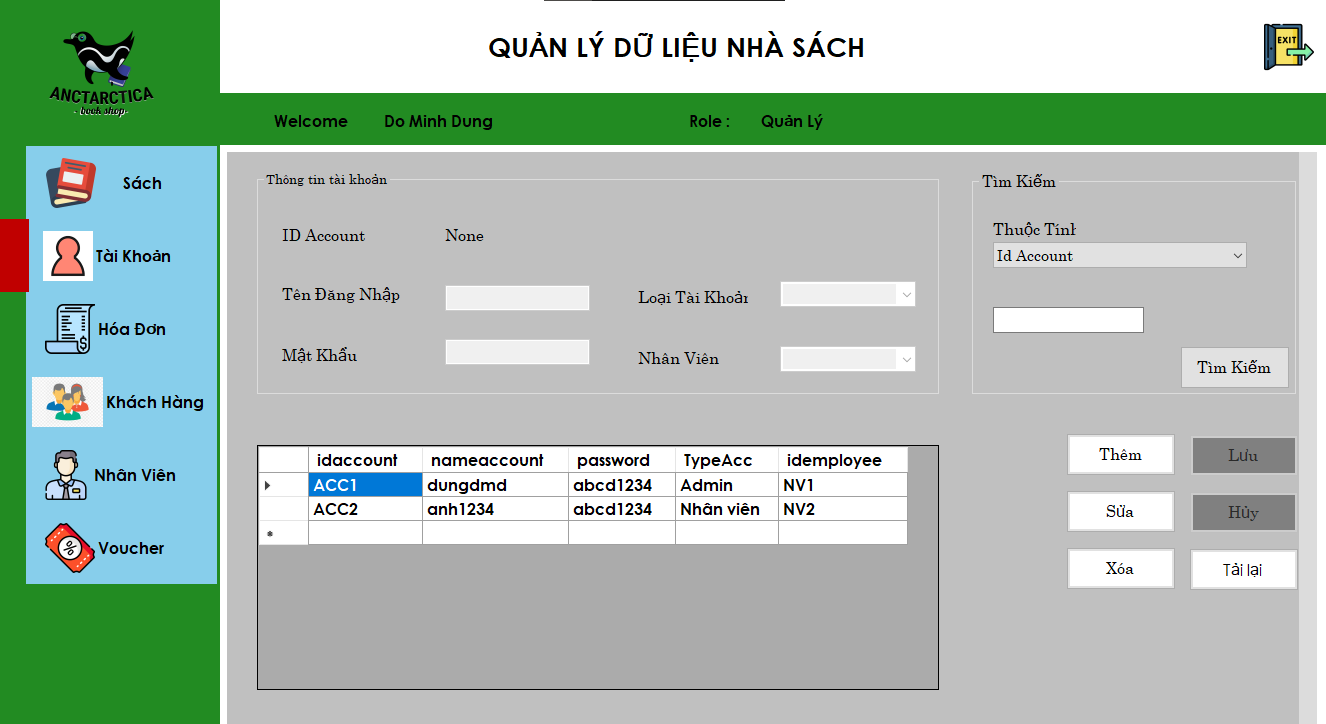


## Giao diện bảng BILLINPUT





## Giao diện bảng ACCOUNT



## Giao diện chức năng thống kê doanh thu



# KẾT LUẬN

## 1. Kết quả đạt được

### 1.1 Về lý thuyết

Nắm vững kiến thức lý thuyết cơ bản phục vụ tốt cho việc thiết kế chương trình. Thiết kế được CSDL tương đối hoàn chỉnh, hiểu và sử dụng được thêm được một số functions…ứng dụng được công nghệ Entity Framework một cách tối ưu, đáp ứng tốt cho việc viết chương trình. Xây dựng được một ứng dụng quản lý cửa hàng sách với đầy đủ các chức năng cần thiết cho việc quản lý bán hàng của người quản lý. Minh họa tốt cho các vấn đề lý thuyết đã trình bày. Củng cố lại các kiến thức đã được học, đặc biệt là kỹ năng phân tích, giải quyết vấn đề. Biết cách áp dụng các kiến thức lý thuyết vào ứng dụng thực tế.

### 1.2 Về đề tài

Sau thời gian nghiên cứu và thực hiện các yêu cầu chính của đề tài, cơ bản đã hoàn tất với những nội dung chủ yếu sau:

- Xây dựng được các chức năng cần thiết cho hệ thống quản lý nhà sách, có giao diện thân thiện, thuận tiện cho người dùng.

- Chức năng cập nhật (thêm, sửa, xóa), tìm kiếm được thực hiện nhanh chóng.

- Biết các sử dụng trigger, function, stored procedure trong hệ quản trị cơ sở dữ liệu và vận dụng vào lập trình Winform, ADO.NET trong C#.

- Hoàn thiện đầy đủ chức năng, giao diện cơ bản của một ứng dụng quản lý để để có thể tiện cho việc quản lý của người chủ cửa hàng như chúng em đã mô tả ở trên, giao diện chương trình thân thiện, dễ sử dụng, trực quan và linh động.

## Hạn chế

Kiến thức thực tế và kiến thức lập trình của chúng em còn hạn chế, nên tính chuyên nghiệp của chương trình chưa cao, một số lỗi còn chưa sửa được, một số tính năng chưa được tối ưu. Chương trình chưa thực sự đầy đủ các tính năng như mong muốn của chúng em và sản phẩm được làm trong thời gian ngắn, kinh nghiệm còn hạn chế nên còn nhiều điểm chưa khắc phục được.

- Khả năng xử lý nghiệp vụ chưa linh hoạt nhạy bén.

- Chưa giải quyết trọn vẹn các vấn đề nảy sinh trong quá trình quản lý

- Một số chức năng của hệ thống chưa hoàn thiện.

## 3. Hướng phát triển

Nghiên cứu sâu hoàn thiện thêm các chức năng của sản phẩm, làm cho phần mềm có ích lĩnh vực quản lý.

Thêm các chức năng như cập nhật video, thông tin về sách. Có thể sử dụng song song bởi nhiều nhân viên và cùng một server.

Chỉnh sửa, thiết kế lại các giao diện và chức năng của ứng dụng cho hoàn chỉnh và mang tính chuyên nghiệp hơn. Xem xét lại các yêu cầu từ khách hàng sử dụng ứng dụng ( người quản lý của cửa hàng). Và sẽ triển khai theo yêu cầu đó. Điều chỉnh lại các mặt hạn chế của đề tài nếu có khả năng sẽ đưa vào thực thi trong thực tế.

# TÀI LIỆU THAM KHẢO

[1] Tài liệu slide giảng dạy, cung cấp bởi GVHD TS. Nguyễn Thành Sơn.

1. Tổng quan về SQL server,

Link: <https://www.dtechvn.com/tong-quan-ve-sql-server.html>, Truy cập ngày 15/11/2021

1. Giới thiệu tổng quan về mô hình 3 lớp chi tiết nhất

Link: <https://bizflycloud.vn/tin-tuc/gioi-thieu-tong-quan-ve-mo-hinh-3-lop-chi-tiet-nhat-20211001162036082.htm,>Truy cập ngày 15/11/2021

1. Một số khái niệm cơ bản về hệ quản trị cơ sở dữ liệu

Link: <https://sites.google.com/site/chiase799/k12_b1>,Truy cập ngày15/11/2022

1. Winform là gì ? Có nên học Winform không ?

Link: <https://plctech.com.vn/winform-la-gi/>, truy cập ngày 15/11/2021